WebSphere. software

IBM

IBM WebSphere Voice Server V5.1.1/V5.1.2 and Avaya Interactive Response V1.3: An Interoperability Guide

- Use Media Resource Control Protocol (MRCP) for interoperability
- Apply best practices to VoiceXML development
- Test and debug VoiceXML applications

James Chamberlain
Bob Anderson
George Clelland
Mukund Chandrasekhar
Steve Sahakyan

Redpaper

ibm.com/redbooks

International Technical Support Organization

IBM WebSphere Voice Server V5.1.1/V5.1.2 and Avaya Interactive Response V1.3: An Interoperability Guide

November 2005

Note: Before using this information and the product it supports, read the information in “Notices” on page v.

First Edition (November 2005)

This edition applies to Versions 5.1.1 and 5.1.2 of WebSphere Voice Server.

© Copyright International Business Machines Corporation 2005. All rights reserved.
Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices v
Trademarks vi

Preface vii
The team that wrote this Redpaper viii
Become a published author ix
Comments welcome. x

Chapter 1. Introduction 1
1.1 Overview of WebSphere Voice Server for Multiplatforms V5.1.x 2
1.1.1 WebSphere Voice Server V5.1.x features 2
1.2 Overview of Avaya IR V1.3 3
1.3 Typical interactive voice response scenario 4
1.4 High level architecture design 5
1.5 Interoperability. 5

Chapter 2. Prerequisites and installation planning 7
2.1 Overview 8
2.2 Architecture review 8
2.2.1 Basic deployment topology 8
2.2.2 Advanced deployment topology 10
2.3 WebSphere Voice Server V5.1 14
2.4 Avaya IR V1.3 15
2.4.1 Hardware requirements. 15
2.4.2 Software requirements for the Avaya IR 16
2.4.3 Avaya IR V1.3 configuration 20
2.4.4 Third-party packages for Avaya IR 37
2.5 Development environment for Avaya IR 40
2.5.1 Avaya Speech Application Builder development prerequisites and planning 40
2.5.2 IBM Voice Toolkit for WebSphere Studio 44

Chapter 3. Developing voice-enabled applications 47
3.1 What is VoiceXML? 48
3.1.1 VoiceXML key benefits 48
3.2 A brief overview of VoiceXML specifications. 48
3.2.1 Elements 48
3.2.2 Common attributes 48
3.3 A Simple VoiceXML Application using Voice Toolkit for WebSphere Studio and Call Flow Builder 49
3.3.1 Creating the Application with Call Flow Builder. 50
3.3.2 Testing the VoiceXML pizza application 55
3.4 Creating a simple VoiceXML application using Avaya SAB V1.3 55
3.4.1 Call flow overview 56
3.4.2 Using the Configurator component 56
3.4.3 Configure and deploy the Application on WebSphere Application Server 61

Chapter 4. Testing and troubleshooting 67
4.1 Testing Avaya IR. 68
4.1.1 Verifying the Installation 68

© Copyright IBM Corp. 2005. All rights reserved.

4.1.2 Checking what packages are installed 70
4.1.3 Establishing telephone connectivity 71
4.1.4 Managing speech resources 72
4.1.5 Speech Synthesis Markup Language test 82
4.2 Load test 84
4.3 Troubleshooting tips 84

Appendix A. Sample code 87
Sample VoiceXML pizza.vxml source code 88
Sample VoiceXML pizzassml.vxml source code 89
Sample Hello World source code from Avaya SAB V1.3 90

Appendix B. Additional material 111
Locating the Web material 111
Using the Web material 111
System requirements for downloading the Web material 111
How to use the Web material 112

Glossary 113

Abbreviations and acronyms 117

Related publications 119
IBM Redbooks 119
Other publications 119
Online resources 119
How to get IBM Redbooks 121
Help from IBM 121

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

© Copyright IBM Corp. 2005. All rights reserved.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX®
DB2®
IBM®
Redbooks™
Redbooks (logo) ™
Tivoli®
WebSphere®

The following terms are trademarks of the Microsoft Corporation in the United States, other countries, or both:

Microsoft®

Enterprise JavaBeans, Java, JavaBeans, JDBC, J2EE, Solaris, Sun, Sun Blade, Sun Fire, Sun Microsystems, Ultra, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Enterprise JavaBeans™
J2EE™
Java™
JavaBeans™
JDBC™
Solaris™
Sun™
Sun Blade™
Sun Fire™
Sun Microsystems™
Sun Solaris™
Ultra™

The following terms are trademarks of other companies:

Avaya and the Avaya logo are trademarks of Avaya, Inc. in the United States, other counties, or both.

NMS is a trademark of NMS Communications Corporation in the United States, other countries, or both.

Gordon Kapes is a trademark of Gordon Kapes, Inc. in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Red Hat is a trademark of Red Hat in the United States, other countries, or both.

SUSE is a trademark of Novell in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Firebird® is a registered trademark of Firebird Foundation Incorporated in the United States, other countries, or both.

The following is a list of W3C terms claimed as a trademark or generic term by MIT, ERCIM, or Keio on behalf of the W3C:

- W3C®, World Wide Web Consortium (registered in numerous countries), HTML (generic), HyperText Markup Language
- HTML (generic), HyperText Markup Language
- HTTP (generic), Hypertext Transfer Protocol
- XML (generic), Extensible Markup Language
- XHTML (generic), The Extensible HyperText Markup Language
- XSL (generic), Extensible Stylesheet Language

Other company, product, and service names may be trademarks or service marks of others.

Preface

The IBM® next generation WebSphere® Voice Server V5.1.1 and V5.1.2 products leverage the power of IBM WebSphere Application Server, IBM's flagship product and leader in the industry. WebSphere Voice Server V5.1 is the first release of a novel (patent pending) architecture for speech servers, based on WebSphere Application Server. No other speech server in the industry can match the on demand capabilities that WebSphere Voice Server is able to provide as a result of its WebSphere Application Server base. Add the power of the Java™ 2 Enterprise Edition (J2EE™) platform, including runtime, services, and container management, and the result is a powerful speech server that integrates into existing IT infrastructures. Having voice on the same network as visual and data applications greatly reduces the initial speech investment. It simplifies the management and achieves improved integration with client content, which is a feat that no other speech vendor can match.

The Avaya Interactive Response (Avaya IR) V1.3 software product delivers next generation voice processing that meets advanced call-routing and voice self-service needs for a contact center enterprise. It utilizes standard NMS hardware for telephony support and a VoiceXML V2.0 compliant browser for application programming interfaces.

IBM WebSphere Voice Server V5.1.1/V5.1.2 and the Avaya Interactive Response V1.3 support full open standards using the:

- Speech Recognition Grammar Specification (SRGS) V1.0 for grammars
- Semantic Interpretation for Speech Recognition (SISR) for semantic interpretation
- Speech Synthesis Markup Language (SSML) V1.0 for text-to-speech markup
- Voice eXtended Markup Language (VoiceXML) V2.0 for speech application markup
- Media Resource Control Protocol (MRCP) V1 Draft 4 for interoperability between the Interactive Voice Response (IVR) and speech server.

This IBM Redpaper is a guide for integrating IBM WebSphere Voice Server V5.1.1/V5.1.2 and Avaya Interactive Response V1.3.

In this Redpaper, we consider best practices as applied to Voice User Interface (VUI) design. We implement and deploy a simple voice-enabled application using Avaya Interactive Response and WebSphere Voice Server V5.1.1/V5.1.2.

We will demonstrate how to develop, test, and deploy a simple voice-enabled application using the VoiceXML markup language. We also demonstrate the use of Automatic Speech Recognition (ASR) and Text-to-Speech (TTS) voice technologies through examples.

With this Redpaper, you can tailor and configure WebSphere Voice Server and Avaya Interactive Response in basic and advanced topologies.

We assume you have a basic knowledge of Interactive Voice Response (IVR) systems and voice-enablement of applications using VoiceXML, Avaya Interactive Response, WebSphere Voice Server, and WebSphere Application Server.

© Copyright IBM Corp. 2005. All rights reserved.

The team that wrote this Redpaper

This Redpaper was produced by a team of specialists from around the world working at the International Technical Support Organization, Raleigh Center.

James Chamberlain is a Senior Software Engineer and certified Senior IT Specialist. He is a project leader at the ITSO, Raleigh Center. He has over 24 years experience in the IT industry and specializes in pervasive computing technologies. His areas of expertise include e-commerce, pervasive computing, portals, AIX®, Linux®, and Java programming. He also architects, designs, and develops solutions using J2EE, XML, Web Services, and IBM software products, including WebSphere and DB2®. Before joining the ITSO, James worked for IBM Global Services on e-commerce system development for IBM Business Partners. He majored in Computer Science at Iowa State University.

Bob Anderson is a Senior Certified I/T Specialist in the USA and is a member of the IBM Certification Board for the Networking discipline. Bob has over 22 years of experience in telecommunications with IBM, starting with IBM Line Switching Marketing and currently working in IBM WebSphere Pervasive software, specializing in conversational access. Bob has written on several topics, including ISDN. Bob is also a Licensed General Class Amateur radio operator.

George Clelland is an Advisory IT Specialist in the EMEA WebSphere Voice Technical Sales team, and is based in the UK. George has worked for IBM for 32 years and has 13 years experience of working with IBM's voice and telephony products. George works extensively with clients advising them on the implementation and usage of IBM voice solutions. He developed much of the early education material for WebSphere Voice Response and gave training classes to clients and IBM in many countries across the world. George has a Bachelor of Science degree in Applied Computing Science from Glasgow University, and is a Member of the British Computing Society.

Mukund Chandrasekhar is a Customer Solution Architect for Real Soft Inc., a leading products and services firm specializing in enterprise-level client self-service solutions. Mukund is based in Cranbury, NJ and has over nine years experience in self-service and contact center industry. His current responsibilities involve designing vendor agnostic OAM&P products, CTI interfaces, and speech enabled self service solutions. He holds a Masters degree in Business Administration from Southern Methodist University, Dallas, and a Masters Degree in Computer Science from University of Texas, Dallas. He is actively involved in the IEEE Princeton, NJ Chapter.

Real Soft Inc's Web site is:

`http://www.realsoftinc.com`

Steve Sahakyan is a pre and post sales consultant at SOFTEL Communications Inc. in North America and is based in Canada. He holds a BA from Concordia University in Montreal, Canada as well as a Masters degree from York University in Toronto, Canada. He has seven years of professional experience in the IT industry and has a vast area of knowledge for both voice and data systems. His areas of expertise include service content management, design and implementation of voice solutions, and IT project management.

SOFTEL's Web site is:

http://www.softel.com

Thanks to the following people for their contributions to this project:

And a specials thanks to our ITSO support staff at the International Technical Support Organization, Raleigh Center:

Margaret Ticknor
Jeanne Tucker
Tamikia Barrow
Linda Robinson

Thanks to our ITSO management:

Jere Cline

And a special thanks to our IBM Pervasive Computing sponsor:

Mary Fisher

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners, and clients.

Your efforts will help increase product acceptance and client satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our papers to be as helpful as possible. Send us your comments about this Redpaper or other Redbooks™ in one of the following ways:

- Use the online **Contact us** review redbook form found at:

  ibm.com/redbooks

- Send your comments in an e-mail to:

  redbook@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HZ8 Building 662
  P.O. Box 12195
  Research Triangle Park, NC 27709-2195

1

Introduction

This chapter provides an introduction to the interoperability of WebSphere Voice Server for Multiplatforms V5.1.x and Avaya Interactive Response V1.3. This chapter contains the following topics:

- Overview of WebSphere Voice Server for Multiplatforms V5.1.x
- Overview of Avaya Interactive Response V1.3
- Typical interactive voice response scenario
- High level architecture design
- Interoperability

© Copyright IBM Corp. 2005. All rights reserved.

1.1 Overview of WebSphere Voice Server for Multiplatforms V5.1.x

WebSphere Voice Server V5.1.x runs as an Enterprise Application in WebSphere Application Server V5.1.1. This extends the WebSphere Application Server benefits of reliability, scalability, and availability to WebSphere Voice Server V5.1.x. These are described in *WebSphere Voice Server for Multiplatforms V5.1.1/V5.1.2 Handbook*, SG24-6447.

In addition, WebSphere Voice Server V5.1.x now supports the industry open standard called Media Resource Control Protocol (MRCP) V1 Draft 4. This is described in 1.5, “Interoperability” on page 5.

1.1.1 WebSphere Voice Server V5.1.x features

At the time of the writing of this Redpaper, the latest release of WebSphere Voice Server is V5.1.2 for Linux and Microsoft® Windows® Server 2003. It is a software product that is part of the larger IBM WebSphere family of products. It can be integrated with other software and hardware telephony products to speech-enable them and their associated applications. The list of features include:

- Support for the following platforms:
  - Microsoft Windows Server 2003
  - Red Hat Enterprise Linux WS/ES/AS for Intel® V3.0 Update 1
  - SUSE Linux Enterprise Server (SLES) V8.0 with Service Pack 2a or 3

  For a list of the current platforms supported, please visit the following URL:

  `http://www.ibm.com/software/pervasive/voice_server/system_requirements`

  To get the latest WebSphere Voice Server fix packs for your operating system, please visit the WebSphere Voice Server Support Page at:

  `http://www-306.ibm.com/software/pervasive/voice_server/support`
- Connectivity to multiple supported IVR vendors that support MRCP V1 Draft 4. For a list of compatible IVRs, please visit the following URL:

  `http://www.ibm.com/software/pervasive/voice_server/ivrgateway.html`
- Support for the following:
  - VoiceXML V2.0 Built-in Grammar Types for all languages
  - Speech Recognition Grammar Specification (SRGS) V1.0

    For more information about SRGS, please visit the following URL:

    `http://www.w3.org/TR/2004/REC-speech-grammar-20040316`
  - Speech Synthesis Markup Language (SSML) V1.0

    For more information about SSML, please visit the following URL:

    `http://www.w3.org/TR/2004/REC-speech-synthesis-20040907`
  - Support for Semantic Interpretation for Speech Recognition (SISR) - W3C Working Draft 1, dated April 2003

    For more information about SISR, please visit the following URL:

    `http://www.w3.org/TR/2003/WD-semantic-interpretation-20030401`
- Grammar-based speech recognition that includes support for inline, built-in grammar, and external grammars referenced via file URI or HTTP URI.

- Capability to barge-in, which allows a user to interrupt the dialog and respond to a prompt.
- WebSphere Voice Server is built upon standards that includes leveraging the power of WebSphere Application Server through J2EE and Enterprise JavaBeans™.
- WebSphere Voice Server leverages the WebSphere Application Server infrastructure for failover, recovery, scalability, and quality.
- WebSphere Voice Server V5.1.x runs as an Enterprise Application in WebSphere Application Server V5.1.1.
- At a Glance System Administration is facilitated by WebSphere Voice Server system administration utilizing the WebSphere Application Server system administrator's console for configuration, administration, troubleshooting, and reviewing log and trace information.
- WebSphere Voice Server V5.1.x has additional voice-specific administration panels to facilitate configuration, monitoring, and troubleshooting.
- WebSphere Edge Server Load Balancer component.

  This allows true load balancing in a round-robin fashion and provides a mean to gracefully take machines out of service for maintenance. It provides automatic failover where the load balancer will automatically remove a non-responding server from service. It also allows you to create and manage a WebSphere Voice Server farm where you can seamlessly add or remove WebSphere Voice Server machines without reconfiguring your IVR.
- WebSphere Application Server Network Deployment.

  This allows a single WebSphere Application Server system administrator's console to manage a cell or grouping of WebSphere Voice Server V5.1.x machines.

Note: For a more detailed description of the IBM WebSphere Voice Server V.5.x, please refer to *WebSphere Voice Server for Multiplatforms V5.1.1/V5.1.2 Handbook*, SG24-6447.

1.2 Overview of Avaya IR V1.3

Avaya Interactive Response (IR) V1.3 is the latest IVR system release from Avaya to deliver a sophisticated platform for the development of advanced client self service solutions.

If you purchased an Avaya IR system that includes hardware and software, the hardware components are usually installed by an Avaya service technician or third-party service provider as part of the product maintenance agreement.

If you purchased a software only system, you are responsible for obtaining the hardware components and operating system (OS) and either installing the hardware platform or arranging the installation through an Avaya service technician or third-party service provider as part of the product maintenance agreement.

In either scenario, the Avaya IR hardware platform (Sun™ Fire V240 server or Sun Blade™ 150 workstation) must be installed on a local area network (LAN) and must have connectivity to a telephony network with which the system will interact for voice communications.

Connectivity to a LAN will also provide connections to a separate server for backup and restore of the system. It will also enable the Avaya IR to interact with optional voice systems, such as, in this case, where the Avaya IR will be interacting with the IBM WebSphere Voice Server V5.1.1/V5.1.2.

Note: The system also requires an external modem for data communications with Avaya services. The system uses this link to send alarms to Avaya and to allow Avaya Services to access your system for maintenance purposes.

In our lab environment, Avaya Services connected to our Avaya IR V1.3 in order to perform an upgrade to our system.

For more in depth information regarding the installation and troubleshooting of Avaya IR V1.3, please refer to the documentation provided at the following URL:

http://www.avayadocs.com

Avaya IR V1.3 uses open standards, such as VoiceXML, SRGS, and SSML, to deliver flexible, standards based, and next-generation voice processing. These standards offer the flexibility to interface with any application server or speech server, such as WebSphere Application Server and WebSphere Voice Server.

To integrate with WebSphere Voice Server, Avaya IR V1.3 uses the Media Resource Control Protocol (MRCP) for Automatic Speech Recognition (ASR) and Text-to-Speech (TTS). Please refer 1.5, “Interoperability” on page 5 for more information.

1.3 Typical interactive voice response scenario

Typically, business applications that are enabled for voice are either simple query or transaction-based applications.

For example, a client can invoke a query application to retrieve information from the database. A simple scenario to demonstrate this would be a call placed to retrieve an account balance, the local weather, movie listings, or stock quotes. This information can now be obtained from a client contact center’s IVR.

These are common examples of query applications that can be enabled for voice. Usually, the application guides the client through a series of menus and forms aided by instructions, prompts, and choices.

The application uses either prerecorded audio files or text-to-speech (TTS), a technology that synthesizes spoken language from text. The client can use spoken words to make selections and provide information. Once the database records are retrieved, the system presents the information by playing back audio files.

On the other hand, a transaction application is used for the purpose of performing specific tasks. Transferring funds between accounts and the buying and selling of stocks are popular examples of this type of application. In the above scenarios, the application starts by authenticating the client through an identification process, and then uses a series of windows to guide the client to enter the required data, such as the source and target accounts for a funds transfer transaction. The system plays instructions, prompts, and available options using prerecorded audio files or by performing text-to-speech. The client responds using spoken commands. Once the required data is collected, the system performs the transaction and returns the result.

This functionality is achieved through the use of VoiceXML applications. For the sake of this Redpaper, we have decided to create a voice application using Avaya’s Studio Application Builder (Avaya SAB) and, alternatively, IBM Voice Toolkit for WebSphere Studio.

For WebSphere Voice Server tools and components, please refer to *WebSphere Voice Server for Multiplatforms V5.1.1/V5.1.2 Handbook*, SG24-6447.

1.4 High level architecture design

Figure 1-1 illustrates a very common setup at a high level.

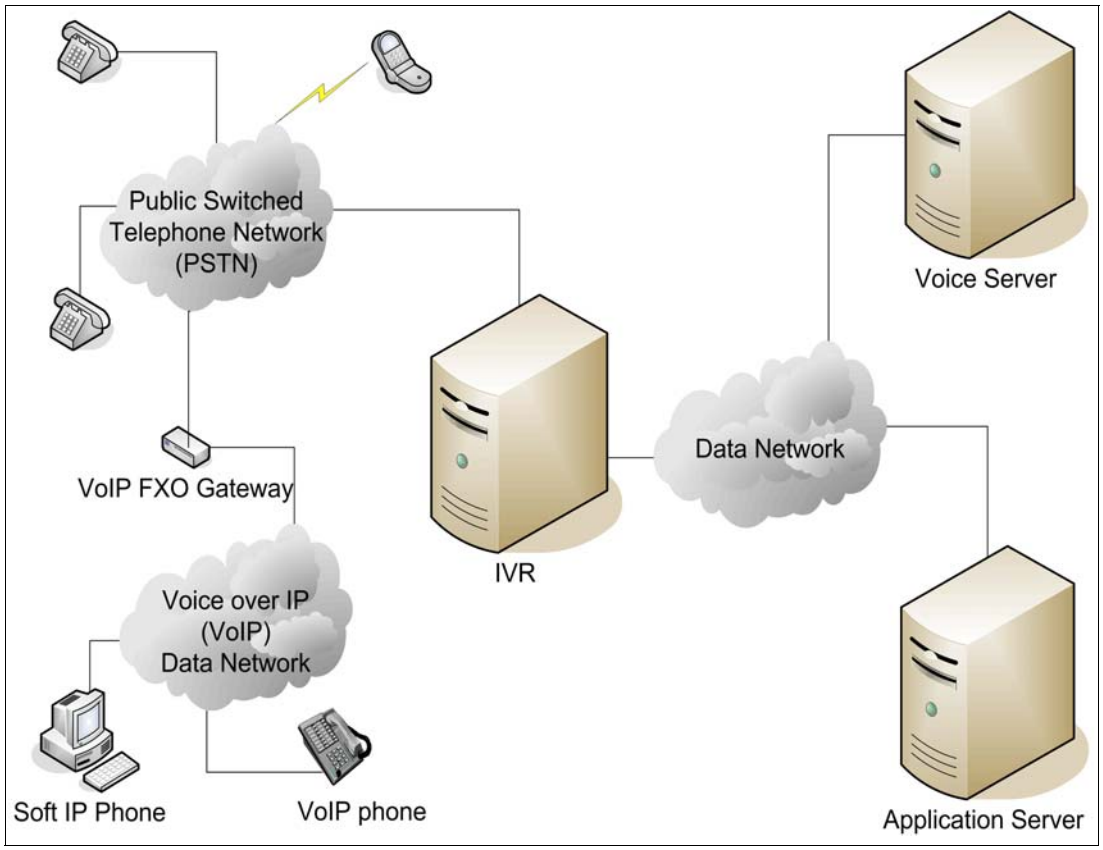

Figure 1-1 High level overview of a typical IVR to voice system setup

1.5 Interoperability

MRCP is defined by the Internet Engineering Task Force (IETF) and the version supported by WebSphere Voice Server is the MRCP V1 Draft 4 specification. MRCP provides the mechanism for controlling speech synthesizer and speech recognizer resources in a standardized manner. This allows any client that is MRCP compliant the ability to talk to any MRCP server.

WebSphere Voice Server V5.1.x can interoperate with Avaya IR V1.3, as they both support the MRCP V1 Draft 4.

For more WebSphere Voice Server Interoperability information, please refer to *WebSphere Voice Server for Multiplatforms V5.1.1/V5.1.2 Handbook*, SG24-6447.

2

Prerequisites and installation planning

This chapter discusses the following:

- Factors to consider in the architecture of a speech platform based on the Avaya IR and WebSphere Voice Server
- The hardware and software prerequisites for each of the products
- The application development environment

© Copyright IBM Corp. 2005. All rights reserved.

2.1 Overview

There are many factors to consider in the design and architecture of a speech platform. However, before embarking on a grand design, it is important to get a good understanding of what functional and environmental requirements must be met by the platform. For example, what will be the call traffic on the platform, how many languages are required, and what level of redundancy and resilience will be required to meet the service level targets? In this chapter, we will look at some these architectural issues and then discuss the hardware and software prerequisites for each of the products.

2.2 Architecture review

In this section, we discuss simple and complex deployments, scalability, and high availability.

2.2.1 Basic deployment topology

A basic deployment would typically consist of a single Avaya IR system, a WebSphere Voice Server system, and a WebSphere Application Server system. This type of configuration would be suitable as a test platform for a larger and more complex production platform, or for a client who does not anticipate a large call volume and has no requirements for resilience. The maximum capacity that this type of configuration could support is dependent on the hardware platform chosen for the Avaya IR and the use made of speech recognition and speech synthesis. As a guide, a mid-range system could be expected to support up to 40 to 48 channels while a high-end system could support on the order of 72 channels. You should always seek advice from your Avaya representative or Avaya Technology and Consulting (ATAC) on the maximum capacity of your system. The limitation of this deployment is that it does not provide any fail-over, recovery, or serviceability. If any of the machines fail or has to be taken out of service for a hardware or software upgrade, the platform will be unable to handle any calls.

Figure 2-1 on page 9 shows the basic deployment that we installed.

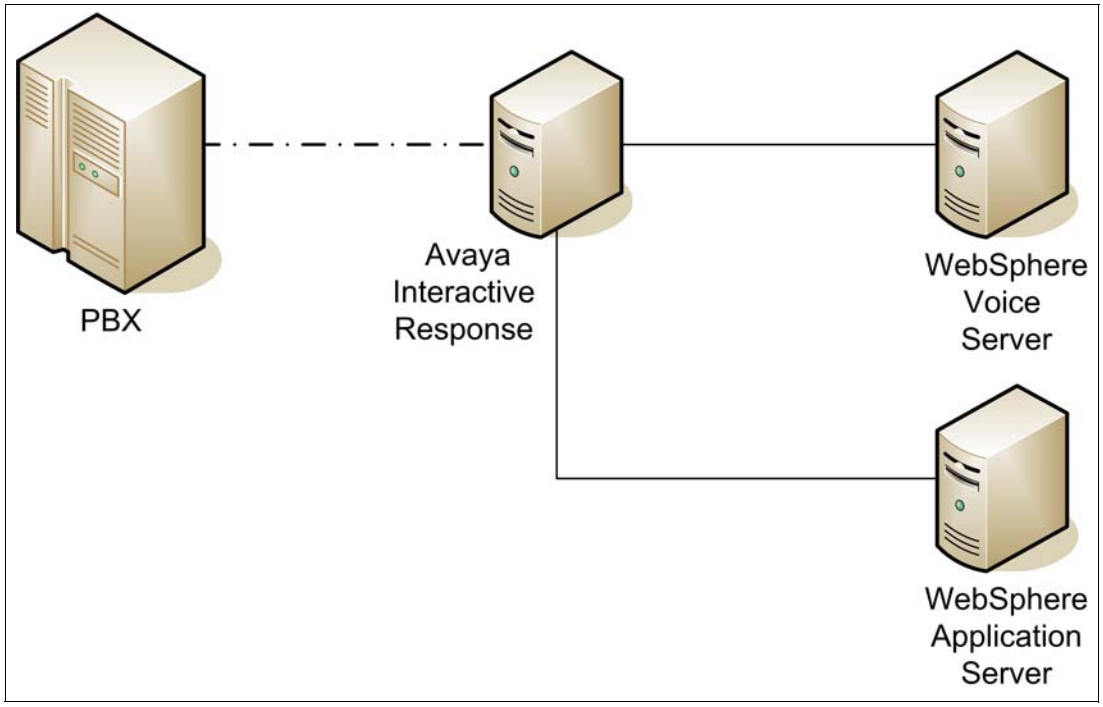

Figure 2-1 Basic deployment topology

In the laboratory, instead of using a PBX, we used a Gordon Kapes System 930 Telephony Simulator to provide our telephony environment. It can support two T1 trunks connected to an IVR and has four analog ports for telephones. In this configuration, we connected one T1 to the Avaya IR using National ISDN 2 (NI2).

2.2.2 Advanced deployment topology

An advanced deployment will be a more common configuration. By *advanced deployment*, we mean a configuration that will typically have more than one Avaya IR system and certainly more than one WebSphere Voice Server system supporting one or more languages, and possibly more than one application server. This type of configuration will be able to handle large numbers of concurrent calls for different applications. A configuration comprised of many systems can be difficult to operate and manage, but to address this issue, we can use the management components of WebSphere Application Server Network Deployment, namely the Deployment Manager and the Load Balancer function of the Edge Components. This type of topology is shown in Figure 2-2.

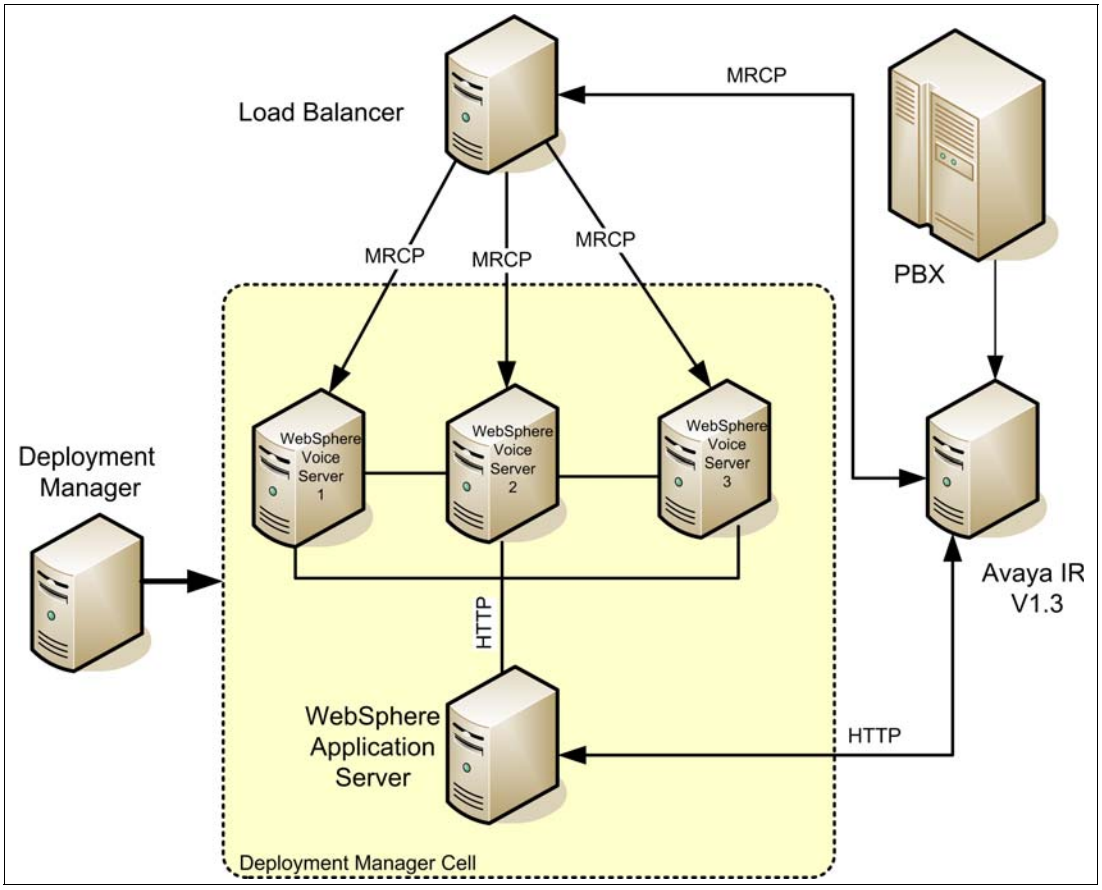

Figure 2-2 Advanced deployment topology showing multiple servers with Load Balancer

In our case, where we have multiple WebSphere Voice Server systems, we only make use of the capability of the Deployment Manager to provide a single point of management for all the systems. This enables the operator to see, at-a-glance, the status of all the WebSphere Voice Server resources and to manage them as required. The Deployment Manager provides many other functions to efficiently manage and operate WebSphere Application Server instances and applications that are distributed across a number of machines.

The Load Balancer function of the Edge Components gives us the capability to cluster together a number of WebSphere Voice Server systems so that, from the perspective of the Avaya IR, they appear to be a single MRCP server resource. The Load Balancer is used to distribute the workload across the WebSphere Voice Server systems. This is an extremely useful capability, as it means that within the cluster of WebSphere Voice Server systems, systems can be taken out of service or fail with no effect on the operation of the platform other than a reduction in capacity. It also means that additional systems can be added to the cluster

to increase capacity without having to change the configuration of, or stop and restart, any of the systems in the platform.

For further reading on the various scenarios and topologies that can be implemented with WebSphere Application Server Network Deployment, we recommend that you refer to the *IBM WebSphere Application Server V5.1 System Management and Configuration WebSphere Handbook Series,* SG24-6195 and the *IBM WebSphere V5.1 Performance, Scalability and High Availability WebSphere Handbook Series,* SG24-6198. There is also more information about these topics, with particular reference to WebSphere Voice Server, in *WebSphere Voice Server for Multiplatforms V5.1.1/V5.1.2 Handbook,* SG24-6447.

Overview of WebSphere Network Deployment Manager

WebSphere Deployment Manager is one of the components of WebSphere Application Server Network Deployment, which is packaged with WebSphere Voice Server V5. The Deployment Manager provides the ability to manage a group of WebSphere Voice Server systems from a single administrative console. The Deployment Manager administrative console has the same look and feel as the administrative console on a stand-alone system.

A Deployment Manager manages a group of systems called a *cell*. Each system in a cell is called a *node*. The nodes in a cell can be on different operating systems. Even though the primary function of the Deployment Manager for a cell of WebSphere Voice Server systems is to provide a centralized point of operation, the Deployment Manager itself does not need to be operational at all times. The WebSphere Voice Server nodes in the cell continue to operate even though the Deployment Manager is stopped. However, you should remember that when a WebSphere Voice Server system is added (federated) to a cell, its own administrative console is removed, and its configuration information is held centrally by the Deployment Manager. Consequently, you should make sure that the configuration information on the Deployment Manager is backed up on a regular basis.

The Deployment Manager communicates with the nodes in its cell through a process called a node agent, which runs on each node. The node agent is responsible, on behalf of the Deployment Manager, for the starting, stopping, synchronization, and configuration of the WebSphere Application Server instance named server1.

Typically, the Deployment Manager is installed on a low-end machine, as it does not require any great amount of resources. You can install it on the same machine as the Load Balancer, which is described in “Overview of WebSphere Edge Server Load Balancer” on page 12.

We would recommend using the Deployment Manager to manage WebSphere Voice Server whenever you have multiple systems installed.

WebSphere Network Deployment Management topology

We set up a multiple machine environment in the ITSO laboratory for the purpose of writing this Redpaper. It consisted of the following units:

- A Gordon Kapes System 930 Telephony Simulator to act as our PBX
- An Avaya IR V1.3 system connected via one T1 trunk running National ISDN 2 (NI2)
- Four WebSphere Voice Server systems, two on Windows Server 2003, one on Red Hat Enterprise Linux, and the fourth on SUSE Linux Enterprise server

  Two systems had US English recognition and text-to-speech, one had UK English, and the other had Canadian French.
- One WebSphere Application Server system on Windows to serve the VoiceXML documents to the Avaya IR

- One Deployment Manager system on SUSE Linux Enterprise server to manage all the WebSphere Voice Server systems
- One Load Balancer system on a SUSE Linux Enterprise server to distribute the load across the two US English systems

This allowed us to manage all the WebSphere Applications Server systems, including WebSphere Voice Server, from a single console. Note that there is not necessarily any relationship between a Deployment Manager cell and a set of multiple servers that have traffic distributed to them through the Load Balancer.

Overview of WebSphere Edge Server Load Balancer

WebSphere Edge Server is a set of components packaged within WebSphere Application Server Network Deployment which, in turn, is packaged with WebSphere Voice Server.

The Load Balancer is used as an IP sprayer to load balance requests across a group of identical WebSphere Voice Server systems. This group is referred to as a *cluster*. It also provides horizontal scalability as nodes (systems) can be added and removed from the group as required in a nondisruptive manner. As a result, a configuration in which a Load Balancer is deployed provides a highly scalable and highly resilient environment. WebSphere Voice Server instances can be stopped and restarted with no impact on the overall platform, other than a reduction in total capacity. In very large environments, a single Load Balancer can work with multiple clusters of WebSphere Voice Server systems. For example, one cluster could be US English and a second cluster could be Latin American Spanish.

The Load Balancer works by representing, to the network, the IP address that is assigned to the cluster. It is this address that is used in the Server Name fields when assigning a Speech Server and TTS Server in the Avaya Speech Proxy Administration. The IP address of the cluster is unique and different from the IP address of the Load Balancer itself and all of the WebSphere Voice Server systems in the cluster. The Load Balancer distributes traffic to each of the systems in the cluster based on weighting factors that are defined in the Load Balancer configuration. Thus, if all the machines in the cluster have the same processing capability, you would set the weights to 1, but if one machine had four processors, another machine had two processors, and a third machine had only one processor, you would set the weights to 4, 2, and 1 respectively, so that the machine which had four processors would be given four times the work of the machine that had only one, and so on.

You need to remember that all the WebSphere Voice Server systems in a Load Balancer cluster must be identical. That is, they must all have the same language installed, and they must have either both ASR and TTS, or ASR only, or TTS only.

Tip: When you set up a Load Balancer cluster, the most important thing to get correct and the most difficult to debug is making sure all the IP addresses and names are correctly defined. In particular, you need to make sure that the IP address of the cluster is correctly aliased on each of the machines in the cluster. This is well documented in the redbook *WebSphere Voice Server for Multiplatforms V5.1.1/V5.1.2 Handbook,* SG24-6447.

Overview of Avaya support for multiple MRCP servers

Avaya IR V1.3 provides an alternative method of implementing a configuration with multiple WebSphere Voice Server systems, which is shown in Figure 2-3. The Avaya IR allows for multiple MRCP servers to be defined for ASR and TTS, and for each server the number of ports or sessions is also defined. During initialization, the Avaya IR MRCP client creates a TCP/IP connection with each of its defined MRCP servers, and then sets up as many Real Time Streaming Protocol (RTSP) sessions with that server as that server has ports defined in the configuration. The MRCP client then allocates and manages the use of the RTSP sessions on a simple round-robin basis. The configuration of MRCP servers is described in 2.4.3, “Avaya IR V1.3 configuration” on page 20.

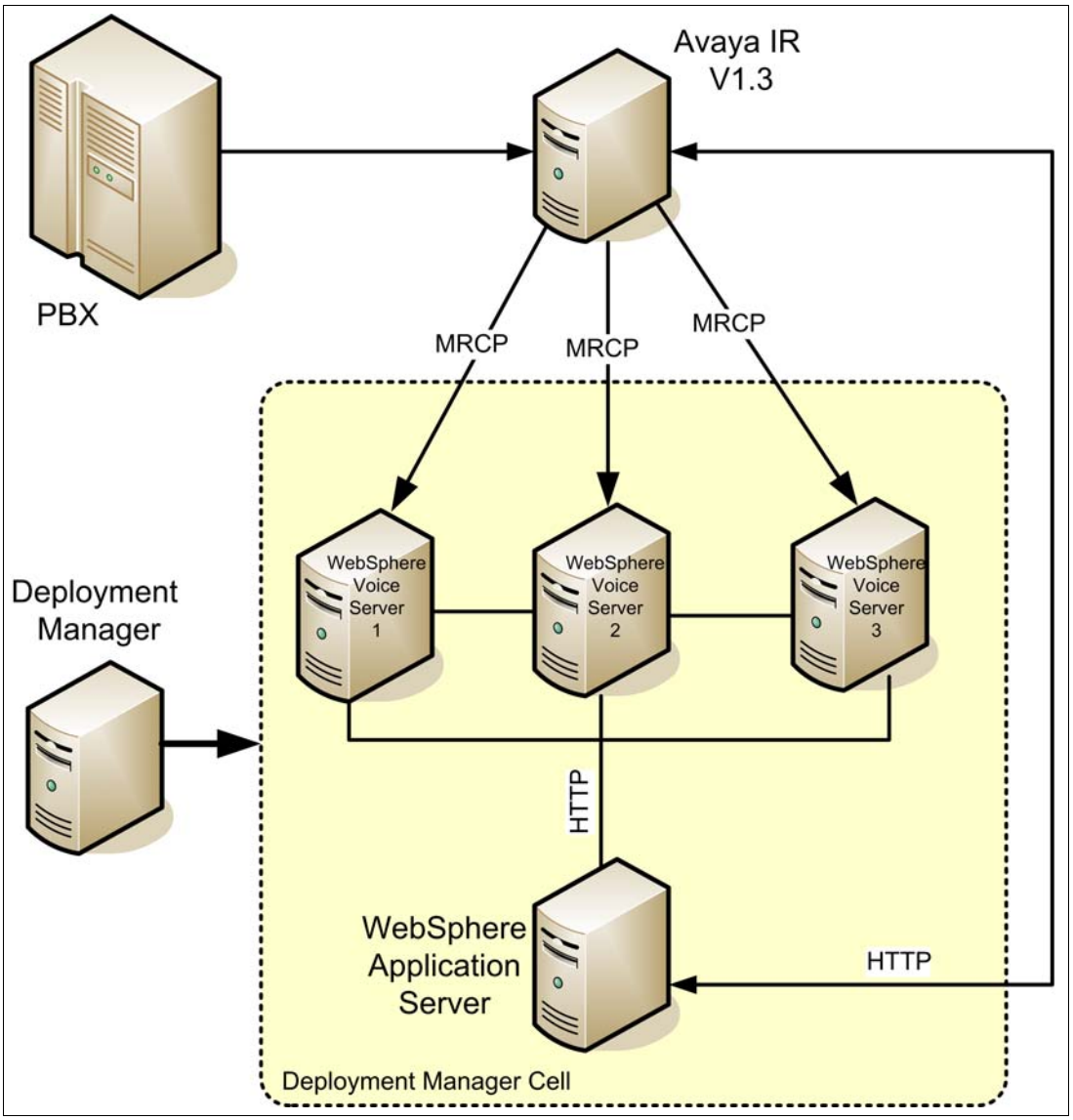

Figure 2-3 Multiple server deployment without Load Balancer

The advantage of this approach is that it is simple to implement and does not require the additional hardware and software needed by the Load Balancer. However, it does not have quite the same flexibility as the Load Balancer, which provides more options in the way the traffic to the WebSphere Voice Server systems is distributed. As well as the simple round-robin method used by the Avaya IR, with the Load Balancer you can weigh the

distribution according to capability, and you can also implement more sophisticated methods based on processor utilization.

For relatively small installations comprising one to two Ayava IVR systems and a similar number of WebSphere Voice Server systems, the simplicity of the Avaya approach may be the most appropriate, but in larger scale installations, we feel the Load Balancer would simplify the administration and operation. For example, if the installation is comprised of 10 Avaya IR systems and 10 WebSphere Voice Server systems and one WebSphere Voice Server system is to be shut down for maintenance, this can be done with one command on the Load Balancer, while with the Avaya approach, the WebSphere Voice Server system would have to be taken out of service on each Avaya IR.

Multiple language considerations

For this Redpaper, we used WebSphere Voice Server V5.1.2. This version supports one language per instance of WebSphere Voice Server. If you need to support ASR and TTS for two or more languages, each unique language has to be installed on separate systems. Avaya IR V1.3 is language neutral as far as VoiceXML applications are concerned. The Avaya MRCP client will pass to the MRCP server the value of the “xml:lang” property from the VoiceXML document processed by the browser. Although Avaya IR V1.3 can define multiple MRCP servers, there is no way to associate one or more of these servers with a specific language.

Attention: With the releases of the products we used, WebSphere Voice Server V5.1.2 and Avaya IR V1.3, if you have to support multiple languages, you will need to use different Avaya IRs for each language, where each of the Avaya IRs is defined to use the specific set of WebSphere Voice Server systems that has that language installed.

In our lab environment, we installed US English and UK English. As we only had a single Avaya IR system, it meant that we had to change the Avaya IT configuration to use the different WebSphere Voice Server systems when we wanted to test each of the languages. In a true production environment, you would need to estimate the proportion of calls which would use each installed language to determine how many systems, and of what capacity, you would require for both Avaya IR and WebSphere Voice Server.

Note: The support of multiple languages on a single WebSphere Voice Server instance is a well understood client requirement. Consequently, although what we have just stated is true for WebSphere Voice Server V5.1.2, you should refer to the latest product documentation for information about the multiple language support in your version. The Information Center for WebSphere Voice Server can be found at:

`http://www.ibm.com/developerworks/websphere/zones/voice/proddoc.html#wvs`

2.3 WebSphere Voice Server V5.1

The prerequisites, planning, and installation procedures for WebSphere Voice Server V5.1 are comprehensively described in the redbook *WebSphere Voice Server for Multiplatforms V5.1.1/V5.1.2 Handbook,* SG24-6447. We used this book and followed its step-by-step procedures for:

- The installation and configuration of WebSphere Voice Server on both Linux platforms and Windows Server 2003
- The application of the WebSphere Application Server Network Deployment V5.1 Fix Pack 1

We also found it useful to refer to the installation instructions and readme files on the WebSphere Application Server Support Web site:

http://www.ibm.com/support/docview.wss?rs=180&uid=swg27004980

The instructions here are quite detailed and should be followed accurately to ensure success.

- The installation and configuration of the Load Balancer
- The installation and configuration of the Deployment Manager

Once we had defined the number of ASR and TTS engines to be started, WebSphere Voice Server worked with the Avaya IR. We did not need to make changes to any other parameters.

2.4 Avaya IR V1.3

The Avaya IR system automates common enterprise interactions and fulfillment between a business and their clients. The Avaya Interactive Response is a standards based speech enabled self service IVR platform that enables enterprises to automate end caller transactions via speech recognition and touch tone applications. Avaya offers a choice of either a complete turnkey hardware, software and application solution or a software only development platform that can be installed on industry standard Sun hardware.

The following sections describe the essential components of an Avaya IR platform.

2.4.1 Hardware requirements

This section covers the hardware requirements of the Avaya IR. A detailed description of the hardware requirements can be found at the following URL:

http://support.avaya.com

Avaya IR's important hardware components are Sun servers running the Solaris™ Operating System and NMS Communication cards. Sun equipment can either be a Sun Fire™ V240 server or a Sun Blade 150 workstation.

The Sun Blade 150 workstation is a low-end machine. The capacity of the Sun Blade 150 workstation is determined by a combination of the connectivity and the application processing needs. For capacity calculations, please contact the Avaya Technology and Consulting Group (ATAC). Your local Avaya sales organization will be able to provide you with the ATAC contact information. The Sun Blade 150 workstation should be configured with the following characteristics:

- CPU: Single 650 MHz UltraSPARC
- Memory: 512 MB
- Disk: Single 80 GB
- Network Interface: Single 10/100 BaseT
- PCI Slots: Three, and two usable for ISDN Interface cards

The Sun Blade 150 workstation supports TDM or VoIP telephony connectivity. For the Avaya IR/IBM WebSphere Voice Server interoperability test, ISDN-NI2 was the protocol chosen. Please refer to the Avaya IR user guide for a complete breakdown of the protocols supported.

Complete specifications for the Sun Blade 150 workstation is available through the SUN Web site at:

http://www.sun.com/desktop/workstation/sunblade150

For large platforms, Avaya recommends a high performance Sun Fire V240 server with the following characteristics:

- CPU: Single 1.34 GHz or dual 1.5 GHz UltraSPARC IIIi processors
- Memory: 8 GB
- Disk: Dual 73 GB/146 GB Hot-Swap Ultra™ SCSI Mirrored disks
- Network Interface: Four -10/100/1000 Gigabit Ethernet
- Power Supply: Dual hot swappable
- PCI Slots: Three, and two are used for ISDN Interface Cards
- 2U rack mountable

Connectivity to the Digital Telephony network (T1 or E1 trunks) is provided via the AG 4000 or AG 4040 NMS Boards (see Figure 2-4). The telephony cards required for the Avaya IR systems are made by Natural Microsystems (NMS):

- The AG 4000/1600 dual cards (required by the Sun Blade 150 workstation)
- The AG 4040 dual card (required by the Sun Blade 150 workstation)
- The AG 4000/3200 quad card (required by the Sun Fire V240 server)
- The AG 4040 quad card (required by the Sun Fire V240 server)

Figure 2-4 NMS AG 4000
Reprinted with permission from NMS

The Avaya IR hardware platforms can have a maximum of two telephony cards. For more information about these cards, see the *AG 4000 Installation and Developer's Manual* or *AG 4040 Installation and Developer's Manual*. This document is available on the Avaya support Web site at the following URL:

`http://support.avaya.com`

or from the NMS Web site at the following URL:

`http://www.nmscommunications.com`

2.4.2 Software requirements for the Avaya IR

The software modules are pre-installed on the Avaya IR. Once the IR has been shipped to a client site, the only thing that needs to be addressed is the configuration and licensing of the IR itself. It is important to understand the key software components on the Avaya IR system, which are listed in the following sections.

Note: To enable the license package on the Avaya IR, the Avaya help desk needs modem access to the IR.

Note: Avaya IR requires the Solaris™ Operating System version to be Solaris 8 Update 8. If you have any other version of the Solaris Operating System, please check with Avaya if it acceptable for the IR.

Packages

Packaged software is a collection of programs created to provide feature functionality on the Avaya IR system. The packages marked *(required)* are the packages you need to have for the Avaya IR to interoperate with the IBM WebSphere Voice Server. The following are the types of packages available:

- Base packages
- Licensed packages
- Optional packages
- Third-party packages

Base package

Base packages include all the basic software that must be installed on every system for core functionality, such as voice system operations and maintenance functions. The following are the components of the base packages:

- Backup and Restore *(required)*: The Backup and Restore Utilities are used to create a full system backup after initial installation that provides system configuration and administration data for disaster recovery.
- Call Bridge Application *(required)*: The Call Bridge Application package allows an application to place an outbound call to a third party and maintain the connection while the caller interacts with the third party.
- JDBC™ Database Interface *(required)*: The JDBC Data Interface Process provides interface to JDBC complaint databases and executes basic SQL commands.
- Interactive Response Base System *(required)*: The Interactive Response Base System package includes the following:
  - PRISM logging and alerting tools
  - Maintenance and initialization processes
  - Voice system and telephony control processes
- Service Creation: The Service Creation Integration package provides the capability to install voice applications on the system that were created by Avaya IVR Designer.
- Universal Call ID *(required)*: The Universal Call ID (UCID) assigns a unique number to a call when it enters that call center network. The single UCID can be passed among platforms, and can be used to compile call-related information across platforms and sites.
- Web Administration *(required)*: The Web Administration component installs the Web Administration feature and System Help. This tool provides a Web interface to administer the IR. Figure 2-5 on page 18 is provided to help understand the capabilities of the tool.

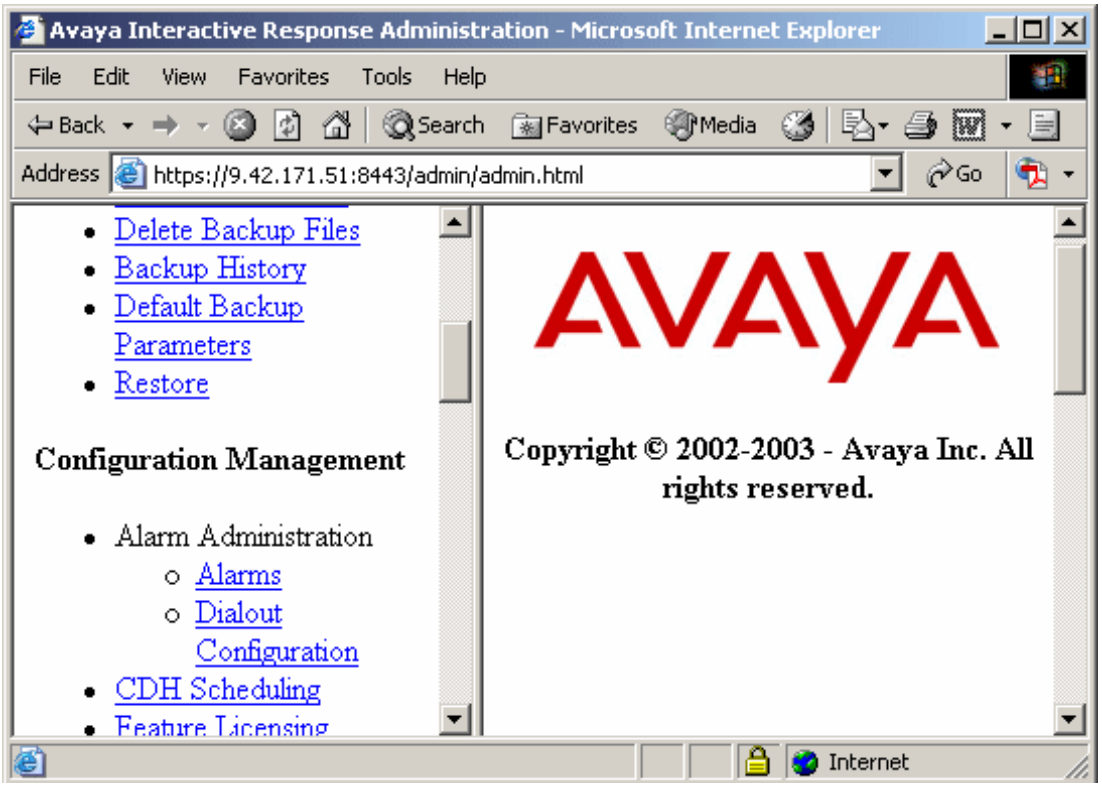

Figure 2-5 Avaya IR Web Administration Tool
Screen capture reprinted by permission from Microsoft Corporation

Licensed package

Licensed packages include the software required to provide digital telephony functions (such as T1/E1 or VoIP) and proxy speech functions (such as Natural Language Speech Recognition and Text-to-Speech). These packages require Right-to-Use (RTU) provisioning on a per-channel basis. The following are licensed packages:

- NMS Package *(required)*: The NMS package implements the digital telephony protocol-level features.
- Proxy Text-to-Speech Package *(required)*: Proxy Text-to-Speech (PTTS) is the proxy server that enables connection to IBM WebSphere Voice Server.
- Speech Proxy Package *(required)*: The speech proxy package includes a speech proxy that enables connection to IBM WebSphere Voice Server.
- Basic Packages required for MRCP: AVsproxy, AVsrproxy, AVttsprxy, AVmrcpasr, and AVmrcptts.

Note: There is often a lot of confusion over the use of the term *Natural Language* in the context of speech recognition. The interpretation of this term varies from vendor to vendor. In its usage here by Avaya, the term means speech recognition technology that is capable of performing recognition on audio input that can be in the form of natural phrases and sentences. The IBM term *Natural Language Understanding* encompasses this capability and extends it to include technology which applies meaning and interpretation to the recognized words within total concept of the application and dialog that is taking place.

Optional packages

Optional packages include the software required for the optional features, such as VoiceXML Interpreter and Avaya Recognizer. These packages require a password to install them on the Avaya IR system and may also require RTU provisioning on a per-channel basis. The following are optional packages:

- VoiceXML Interpreter *(required)*: The VoiceXML Interpreter package installs the Avaya Voice Browser (AVB) software. The Avaya IR V1.3 supports a VoiceXML V2.0 compliant voice browser as well as a VoiceXML V1.0 compliant browser.
- ASAI *(required)*: The Adjunct Switch Application Interface (ASAI) provides a local area network interface between PBX and adjuncts.
- CTI DIP: Applications that make function calls to the Computer Telephony Integration (CTI) data integration process (CTI DIP) on the Avaya IR system will need this package.
- Fax Package: Supports fax.
- PDS Integration: The Predictive Dialing System (PDS) Integration package is software that integrates the functionality of the Avaya IR system and the Avaya Predictive Dialing System

Third party packages for Avaya IR

There is Avaya IR OEMs software from three different Independent Software Vendors (ISVs). They are Real Soft Inc, Cleo, and Gold Systems, which provide additional features and functionality for the IR. These packages are described in 2.4.4, “Third-party packages for Avaya IR” on page 37.

Note: Instructions on how to list the packages installed on your Avaya IR is provided in 4.1.2, “Checking what packages are installed” on page 70. The output of this command, which shows the packages on our system, is in a UNIX® text file called pkginfo.txt that you can extract from the downloadable REDP3977.zip file. Appendix B, “Additional material” on page 111 contains additional information about how to download the REDP3977.zip file and describes its contents.

Validate the installation

It is time to validate the software installed on the IR. Using the Web Admin’s Feature Licensing link, it is possible to find what features are licensed to run on the Avaya IR, as shown in Figure 2-6 on page 20.

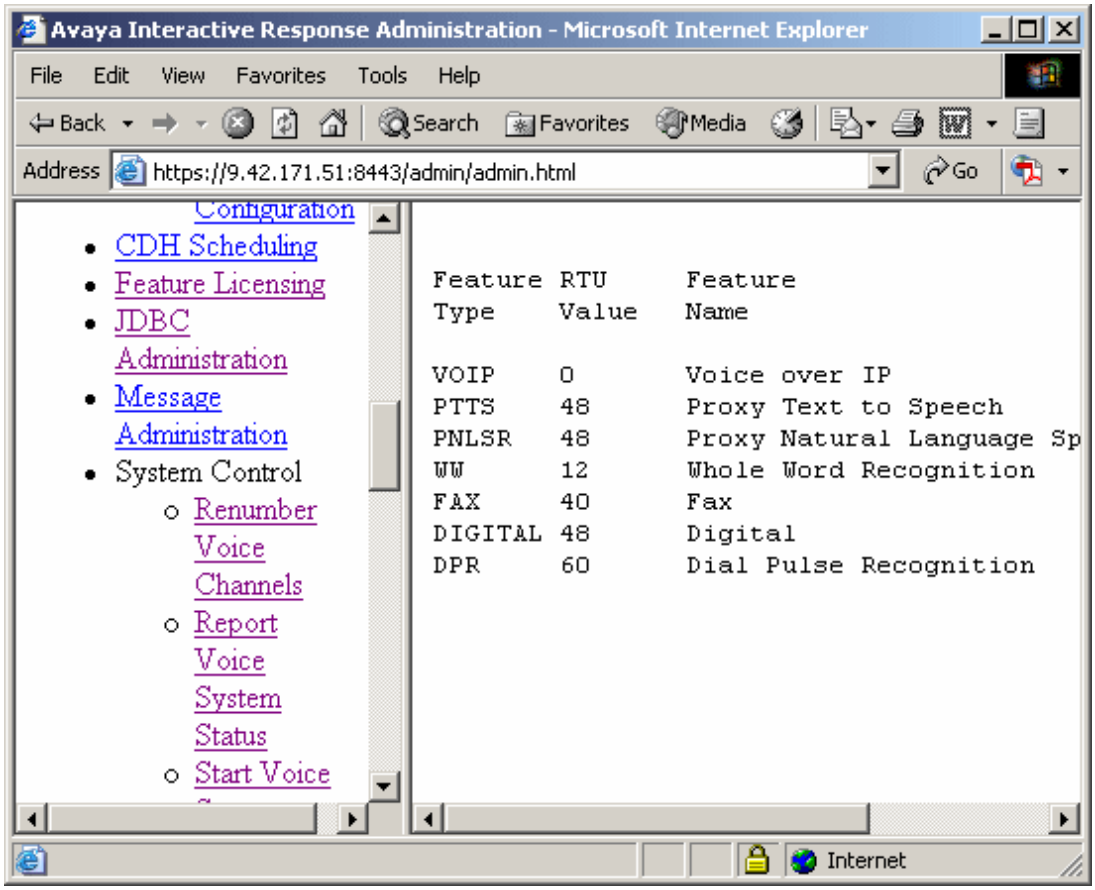

Figure 2-6 View the features licensed on the Avaya IR
Screen capture reprinted by permission from Microsoft Corporation

2.4.3 Avaya IR V1.3 configuration

In this section, we describe how to configure and manage speech resources on the Avaya IR with respect to WebSphere Voice Server using the Avaya Web Administration interface. For information about other aspects of configuring and managing the Avaya IR, for example, configuring the telephony protocols, you should refer to the Avaya IR product documentation.

On the Avaya IR, the MRCP servers for speech recognition and speech synthesis are defined separately. The reference document for this procedure is *Avaya Interactive Response Release 1.3 MRCP Feature.* At the time of writing, this document had not been assigned a document number.

Configuring speech recognition servers

1. Launch and log in to the Avaya Web Administration tool. From the navigation panel on the left, click **Administration** from the Speech and DPR Administration group. The Speech Proxy Administration panel is displayed, as shown in Figure 2-7 on page 21. This panel is the starting point for configuring everything related to speech recognition and TTS servers.

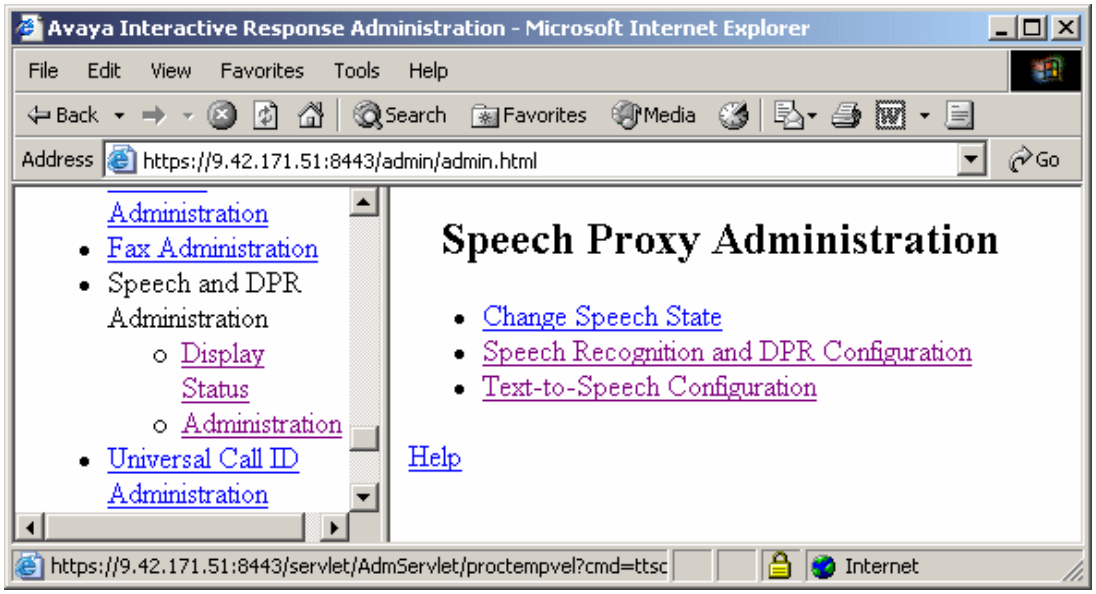

Figure 2-7 Speech Proxy Administration window
Screen capture reprinted by permission from Microsoft Corporation

2. To define a speech recognition server, click **Speech Recognition and DPR Configuration**, which will take you to the Speech Recognition and DPR Configuration window. If no servers have been defined, the window in Figure 2-8 is displayed; otherwise, the window in Figure 2-10 on page 23 is displayed.

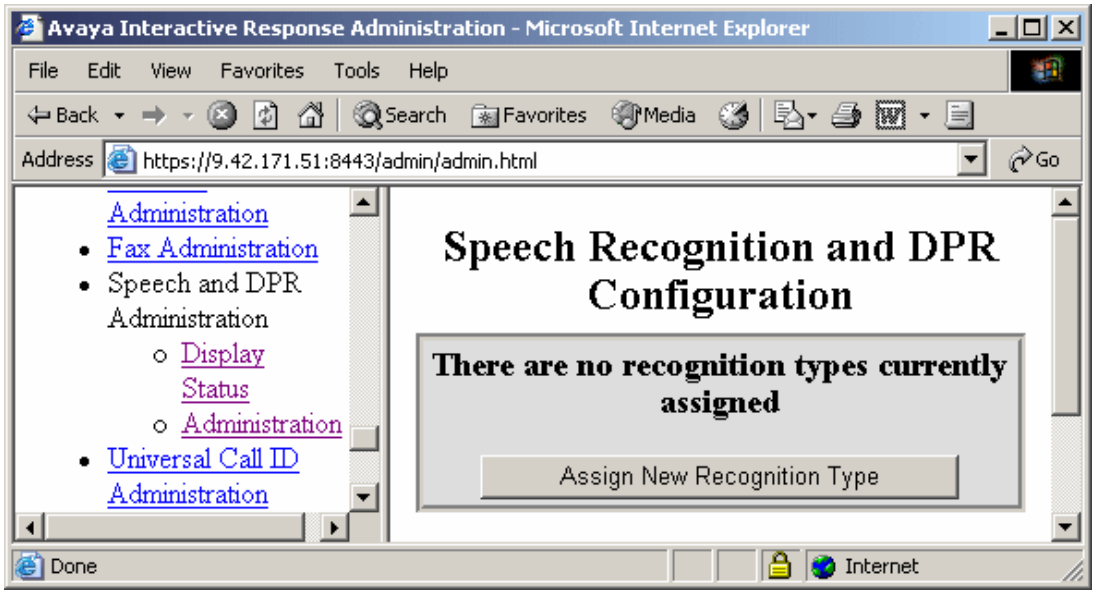

Figure 2-8 Speech Recognition and DPR Configuration: nothing assigned
Screen capture reprinted by permission from Microsoft Corporation

For WebSphere Voice Server, we need to define a recognition type of OPSR4. Click **Assign New Recognition Type** to display the Assign Speech Recognition or DPR type window.

Note: Dial Pulse Recognition (DPR) is a feature that allows the voice response system to recognize digits dialed by callers using rotary dial telephones.

3. On the Assign Speech Recognition or DPR Type window, select **OSPR4** as the Recognition Type, and select **mrcp** as the Engine, as shown in Figure 2-9, and then click **Submit** to make the assignment. The system now displays a window as shown in Figure 2-10 on page 23. Notice the warning message “The change succeeded, please stop and restart the voice system.” that is displayed underneath the window title. This message is displayed any time you make a change to the configuration. This means that, for the change to become effective, you must stop the voice system and restart it.

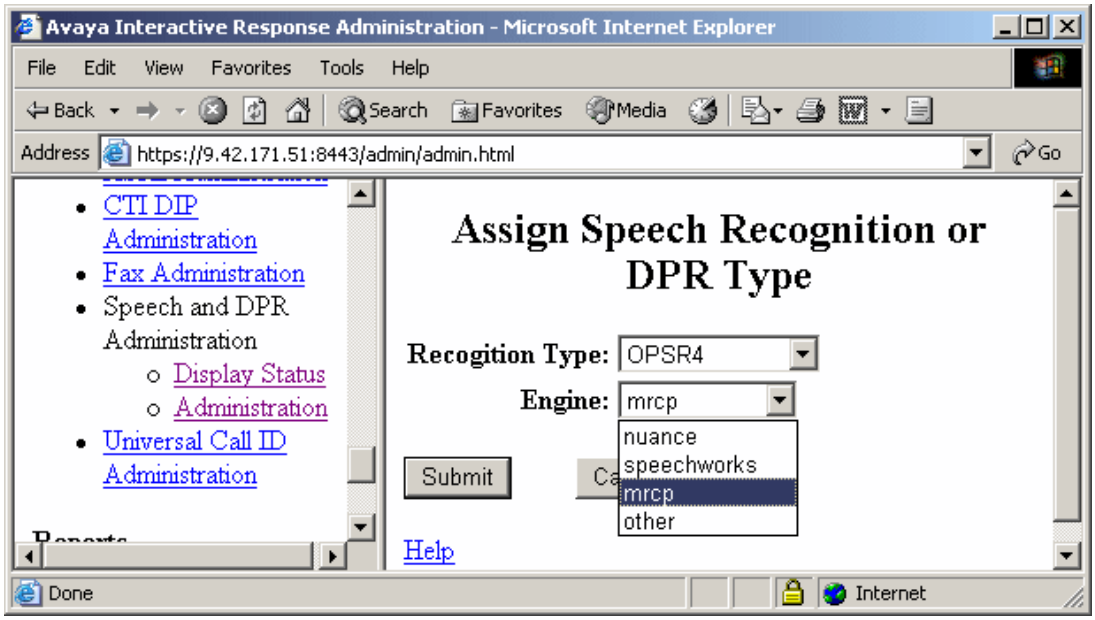

Figure 2-9 Assign Speech recognition or DPR Type

Screen capture reprinted by permission from Microsoft Corporation

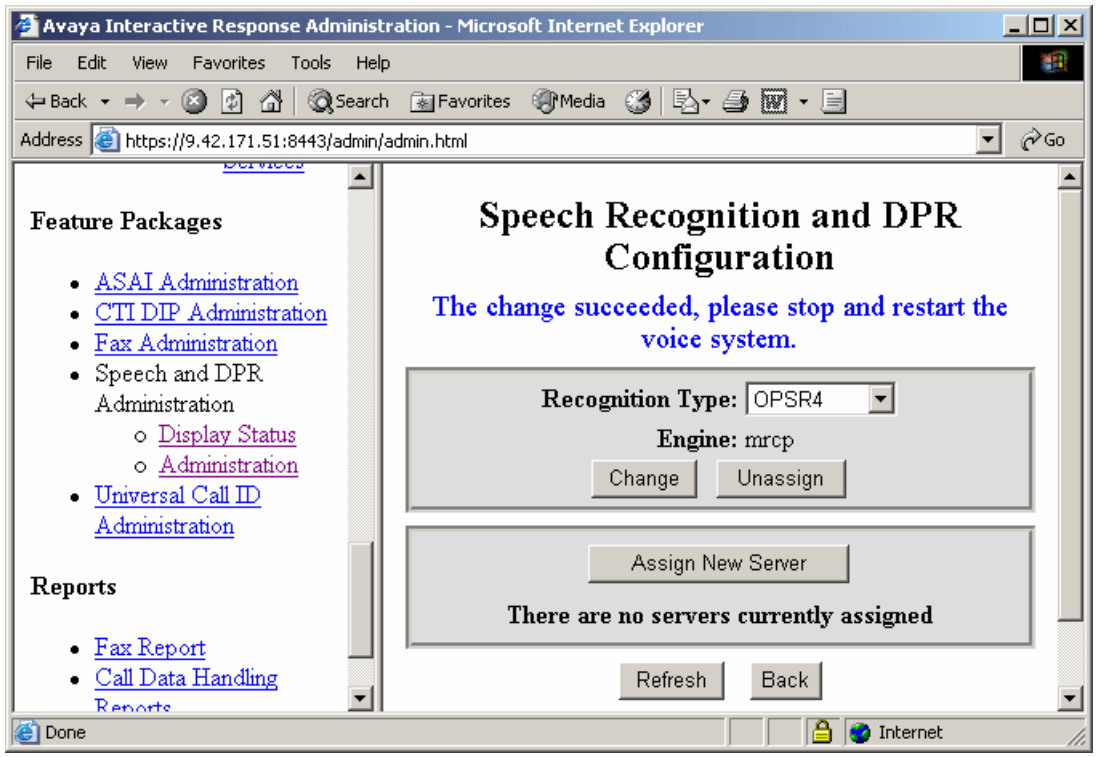

Figure 2-10 Speech Recognition and DPR Configuration: no servers assigned
Screen capture reprinted by permission from Microsoft Corporation

4. Click **Assign New Server** to bring up the Assign Speech Recognition or DPR Server window, as shown in Figure 2-11 on page 24.

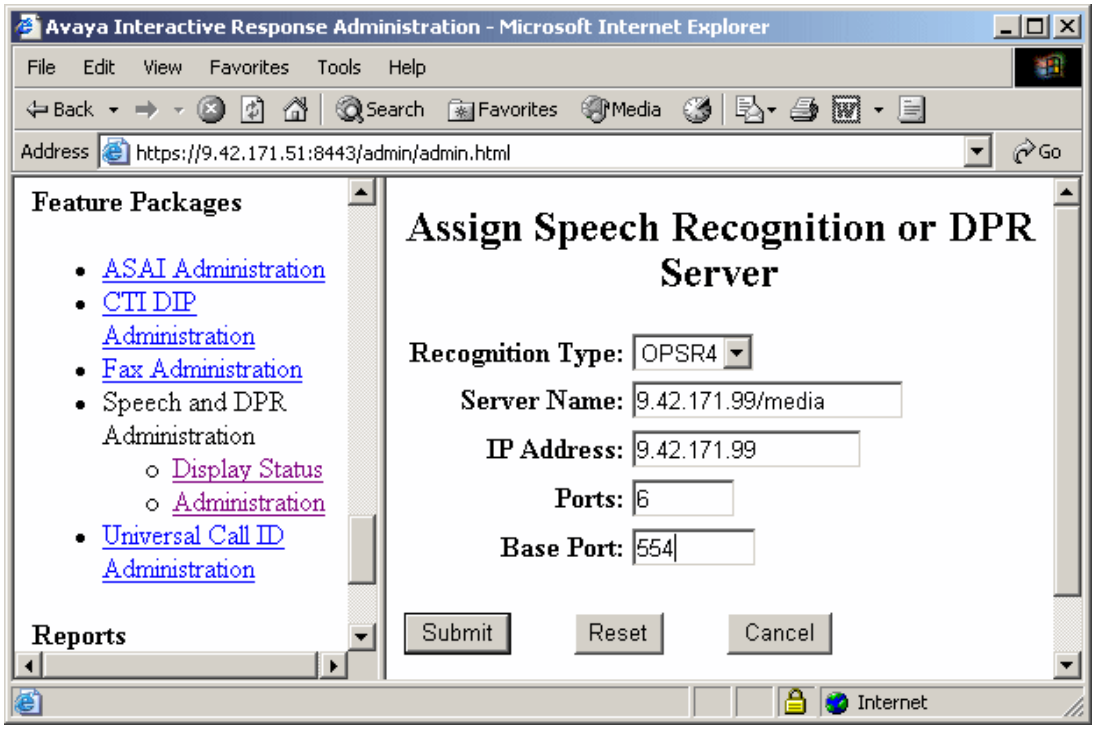

Figure 2-11 Assign Speech Recognition or DPR Server window
Screen capture reprinted by permission from Microsoft Corporation

Complete the fields in this window using the information in Table 2-1. Click **Submit** to change the server definition and return to the Speech recognition and DPR Configuration window, which displays details of all the defined servers.

Table 2-1 Speech Recognition and DPR Configuration fields

Field	Description
Recognition Type	This is preset to OPSR4 from the previous window. Do not change it.
Server Name	The name to be assigned to this server. This must be of the form *name*/media, where name can be an IP address or a simple or fully qualified host name.
IP Address	The IP address of the server. If you are using a cluster of WebSphere Voice Server systems with the Load Balancer, this should be the IP address of the cluster.
Ports	The number of recognition engines on the server. If you are using a cluster of WebSphere Voice Server systems with the Load Balancer, this should be the total number of recognition engines in the cluster.
Base Port	This the port number being used for communication with the server. The well known port for MRCP is 554, so you should enter the value 554, unless you have chosen to use a different port number and made corresponding changes to WebSphere Voice Server.

The Avaya IR V1.3 allows multiple servers to be defined for a recognition type. If you need more than one WebSphere Voice Server system to satisfy your recognition requirements, then you can either define multiple servers to Avaya or use the Load Balancer. Refer to 2.2.2, “Advanced deployment topology” on page 10 for a discussion on these different approaches to scalability.

Figure 2-12 shows the Speech Recognition and DPR Configuration window with a single server, which could be a Load Balancer cluster, and Figure 2-13 on page 26 shows two servers defined.

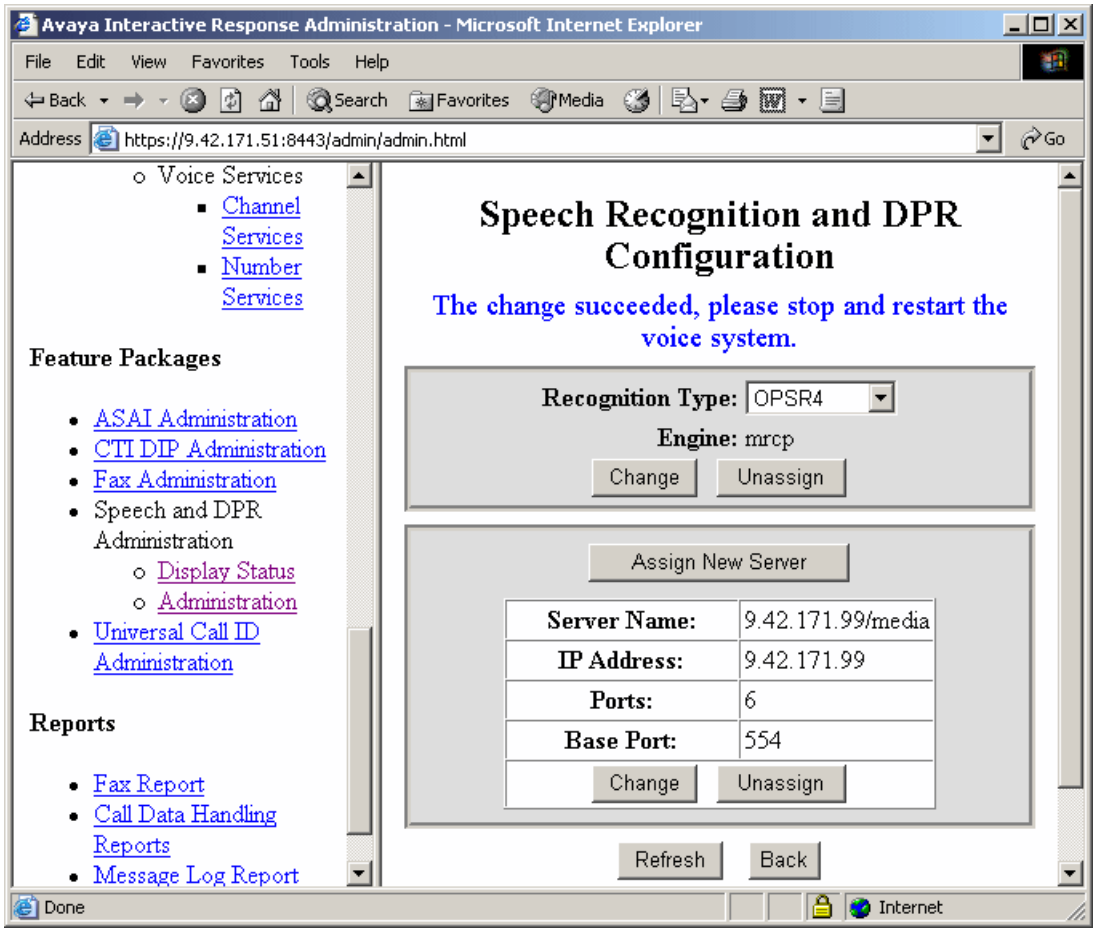

Figure 2-12 Speech recognition and DPR Configuration: single server
Screen capture reprinted by permission from Microsoft Corporation

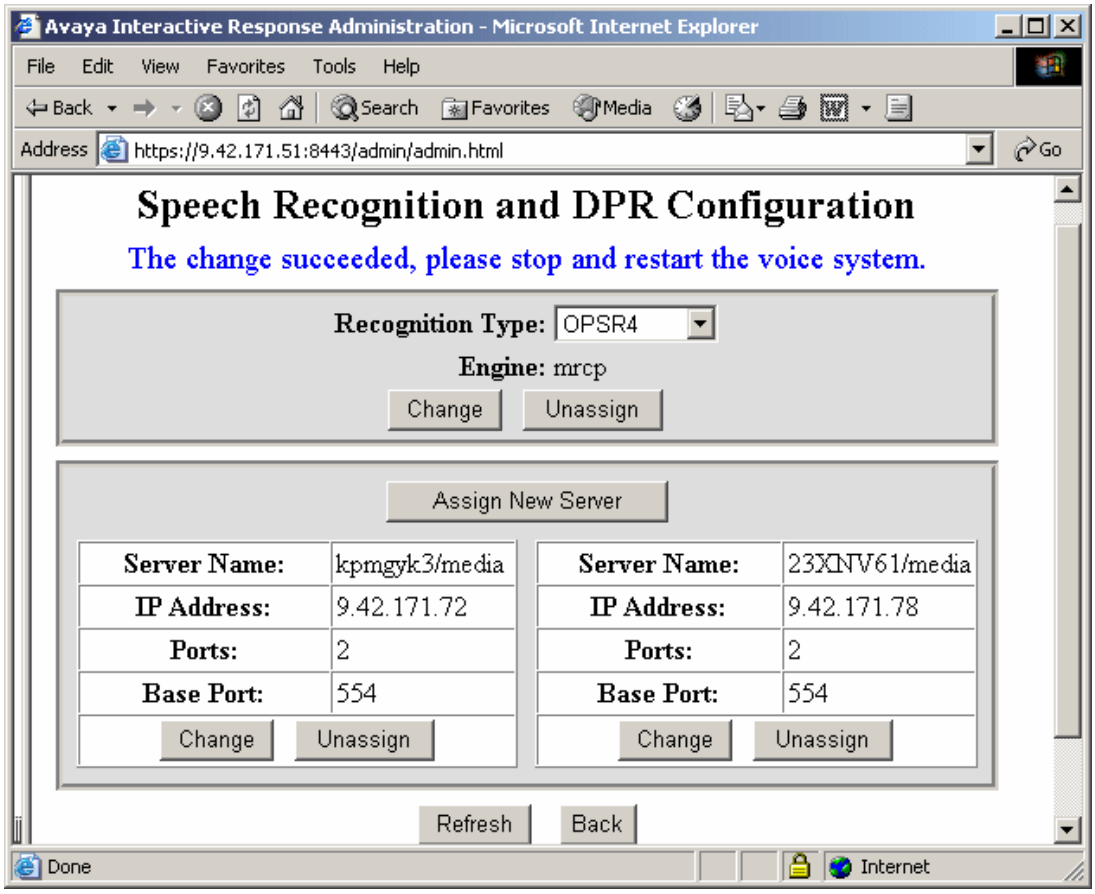

Figure 2-13 Speech Recognition and DPR Configuration: multiple servers
Screen capture reprinted by permission from Microsoft Corporation

You have now completed the configuration of the speech recognition servers. If you have TTS servers to configure, this task is described in “Configuring TTS servers” on page 27, or if you have no more speech resources to define, you must stop and restart the Voice System using the Web Administration tool to make the configuration effective.

Configuring TTS servers

The procedure for defining TTS servers is very similar to the procedure for defining speech recognition servers, as you first specify the TTS Type and then assign one or more servers to that type. The steps are as follows:

1. On the Speech Proxy Administration window (Figure 2-7 on page 21), click **Text-to-Speech Configuration** to get the Text-to-Speech Configuration window. If there are no TTS types assigned, the window shown in Figure 2-14 is displayed. Our screen capture was taken after a TTS type and servers had been defined and then removed, so it shows the Default Voice value that had been assigned. If no TTS types or servers have previously been defined, then the Default Voice value will be blank. On this window, click **Assign New Text-to-Speech Type**.

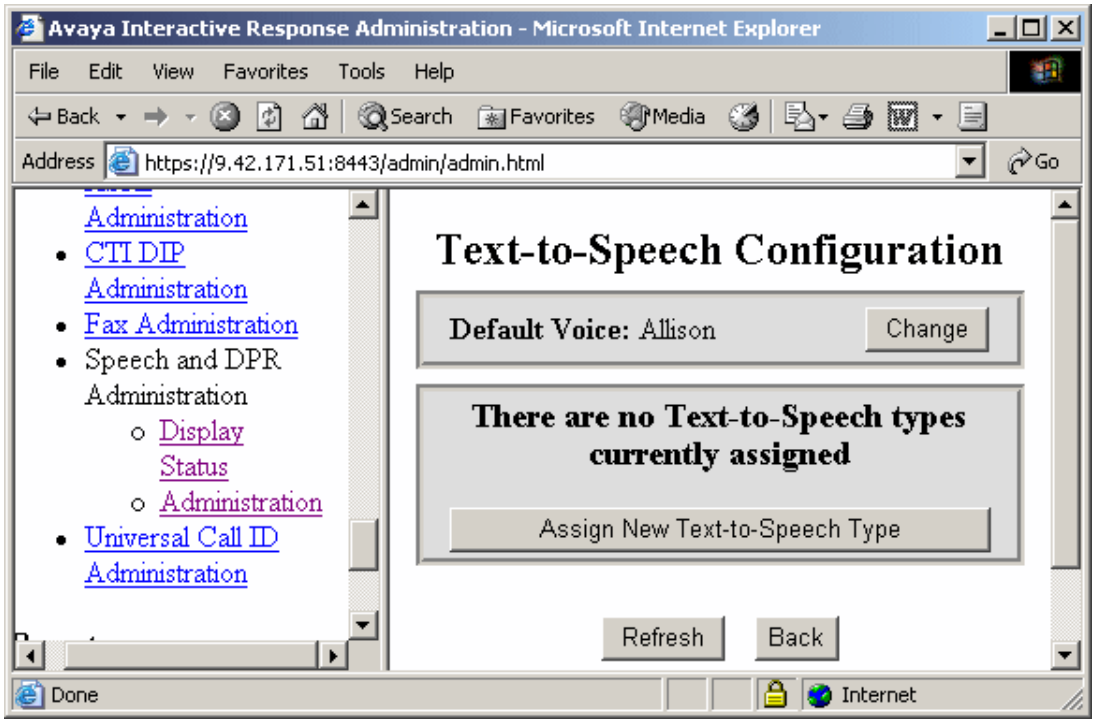

Figure 2-14 Text-to-Speech Configuration window: no Types assigned

Screen capture reprinted by permission from Microsoft Corporation

2. The next window is the Assign Text-to-Speech Configuration window. In this window, select a Text-to-Speech Type from the menu list, and select **MRCP** from the Engine menu list. Figure 2-15 shows the selections we made. The Avaya IR can only have one Text-to-Speech Type if it is using MRCP for the connection to the TTS server. We selected **TTS0** as the Text-to-Speech Type, as it is the first one in the list. Click **Submit** to make the assignment.

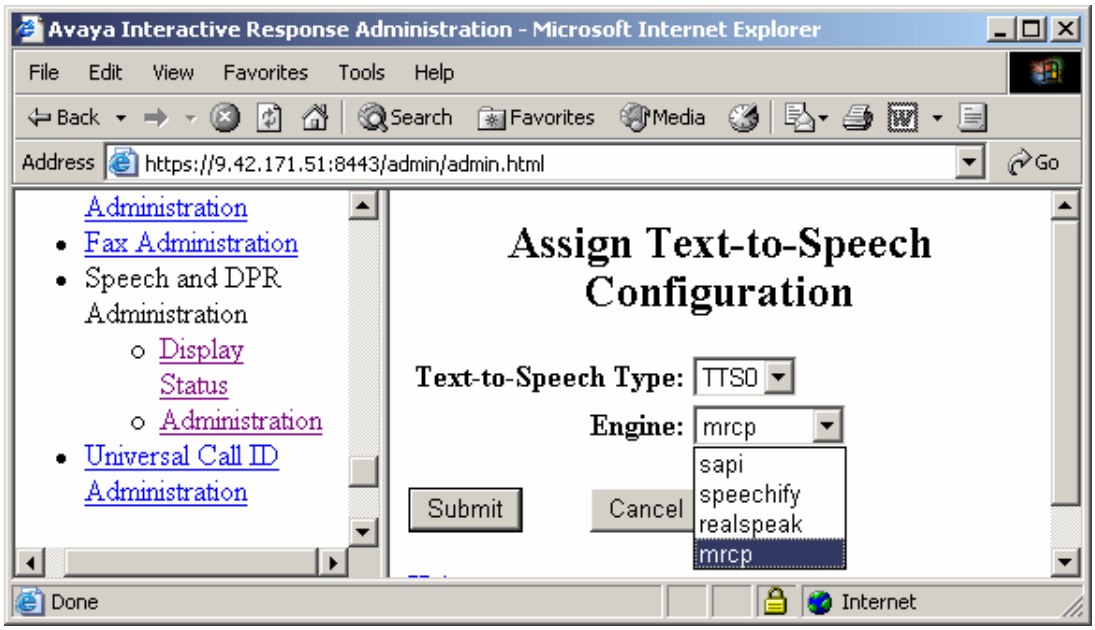

Figure 2-15 Assign Text-to-Speech Configuration
Screen capture reprinted by permission from Microsoft Corporation

The system now displays the window shown in Figure 2-16 on page 29. Notice the warning message displayed at the top of the window, which you will see any time you make a change to the configuration. This means that for the change to become effective, you must stop the voice system and restart it.

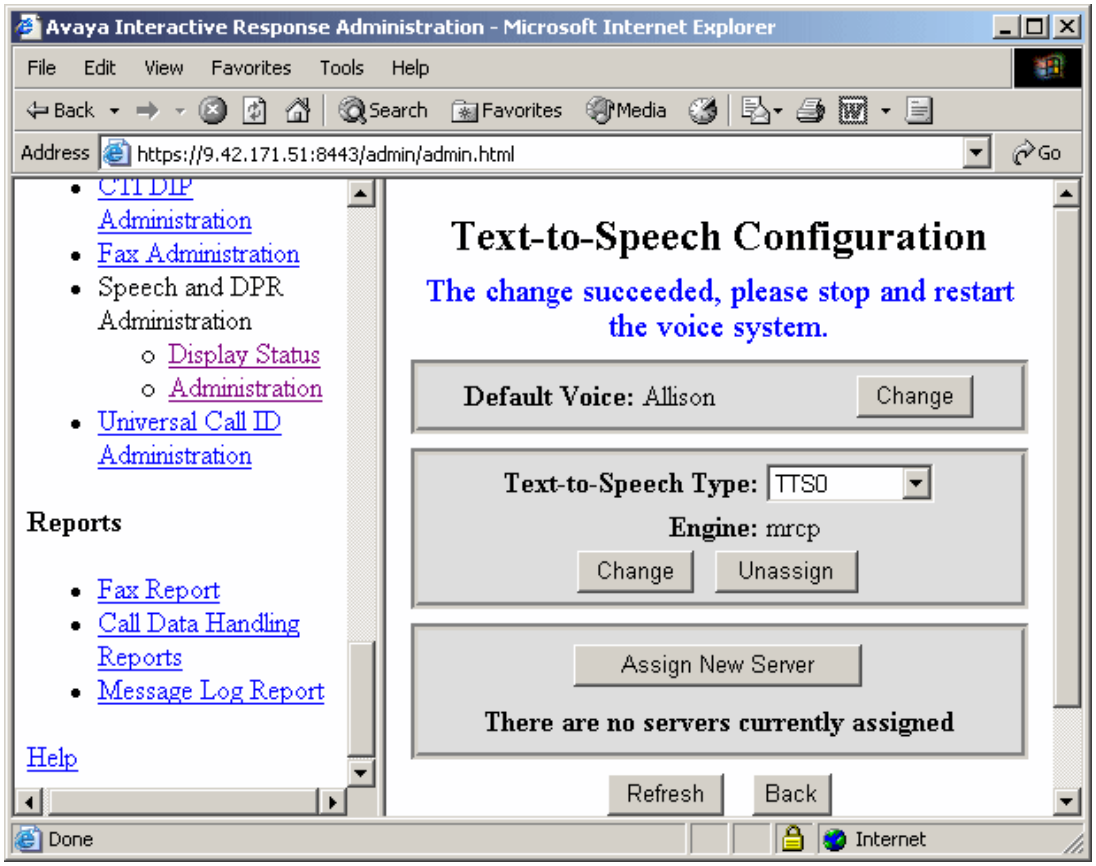

Figure 2-16 Text-to-Speech Configuration: no servers assigned
Screen capture reprinted by permission from Microsoft Corporation

3. The next step is to assign the TTS servers. Click **Assign New Server** to bring up the Assign Text-to-Speech Server window. Complete the fields in this window using the information in Table 2-2. Click **Submit** to change the server definition and return to the Text-to-Speech Configuration window, which displays details of all the defined servers.

Table 2-2 Text-to-Speech Configuration fields

Field	Description
Text-to Speech Type	This is preset to the value from the previous window. Do not change it.
Server Name	The name to be assigned to this server. This must be of the form *name*/media, where name can be an IP address or a simple or fully qualified host name.
IP Address	The IP address of the server. If you are using a cluster of WebSphere Voice Server systems with the Load Balancer, this should be the IP address of the cluster.
Ports	The number of text-to-speech engines on the server. If you are using a cluster of WebSphere Voice Server systems with the Load Balancer, this should be the total number of text-to-speech engines in the cluster.
Base Port	This the port number being used for communication with the server. The well known port for MRCP is 554, so you should enter the value 554, unless you have chosen to use a different port number and made corresponding changes to WebSphere Voice Server.

The Avaya IR allows multiple servers to be defined for a TTS type. If you need more than one WebSphere Voice Server system to satisfy your recognition requirements, then you can either define multiple servers to Avaya or use the Load Balancer. Figure 2-17 shows the Text-to-Speech Configuration window with a single server, which could be a Load Balancer cluster, and Figure 2-18 on page 31 shows two servers defined.

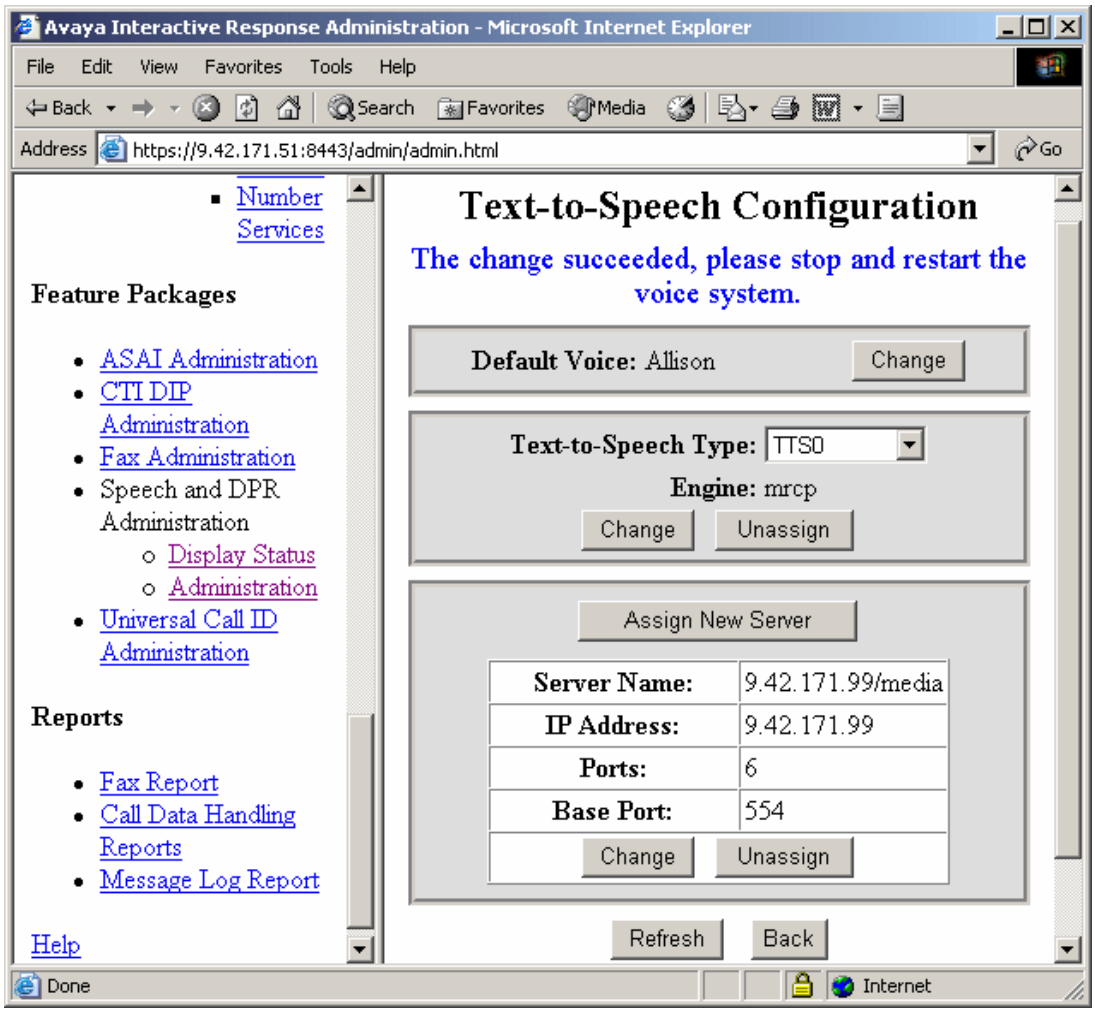

Figure 2-17 Text-to-Speech Configuration: single server
Screen capture reprinted by permission from Microsoft Corporation

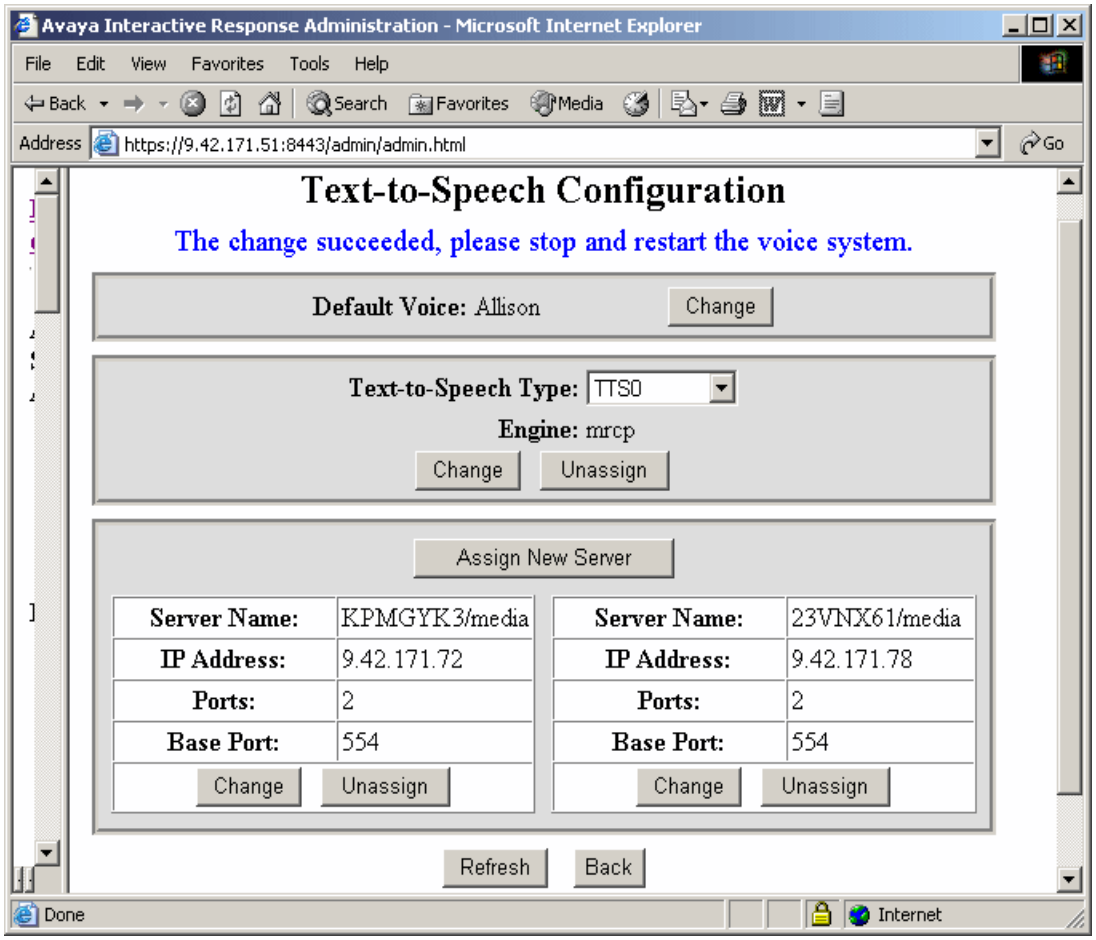

Figure 2-18 Text-to-Speech Configuration: multiple servers
Screen capture reprinted by permission from Microsoft Corporation

4. From the Text-to-Speech Configuration window, you can set the Default Voice value. Click **Change** on the Default Voice bar. The Text-to-Speech Change System Parameters window appears, as shown in Figure 2-19. Enter the name of the voice you want to use as the Default Voice. The names of the voices you have installed can be found by referring to the TTS Configuration parameters on the WebSphere Administration Console. Click **Submit** to set the value of the Default Voice.

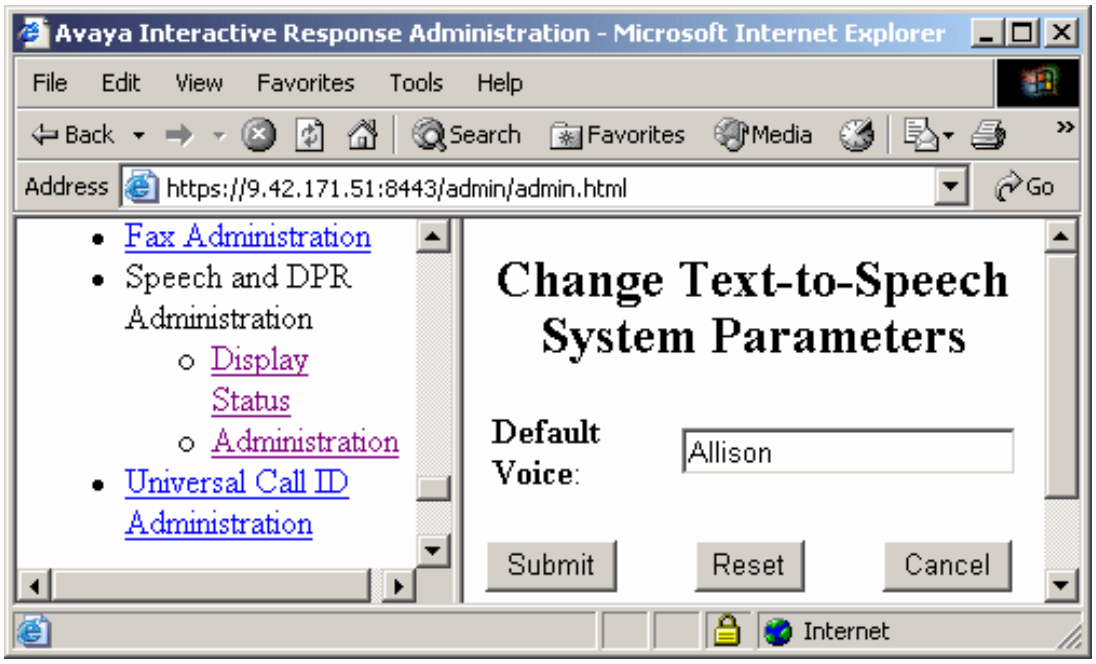

Figure 2-19 Change Text-to-Speech System Parameters
Screen capture reprinted by permission from Microsoft Corporation

You have now completed the configuration of the TTS servers. If you are finished making changes to the configuration, you must now stop and restart the Voice System using the Web Administration tool to make the changes effective.

Configuring application URIs

In order to have a VoiceXML application handle a call, you need to assign the Universal Resource Identifier (URI) which references the application either to a channel or a called number or a calling number.

Using Channel Services

To assign a URI to one or more channels, click **Channel Services** in the navigation panel to the left of the Web Administration window. The Channel Service window will be displayed, showing the current channel assignments. The window initially shows the first 10 channels, as shown in Figure 2-20 on page 33. The number of channels displayed can be altered using the Display menu list.

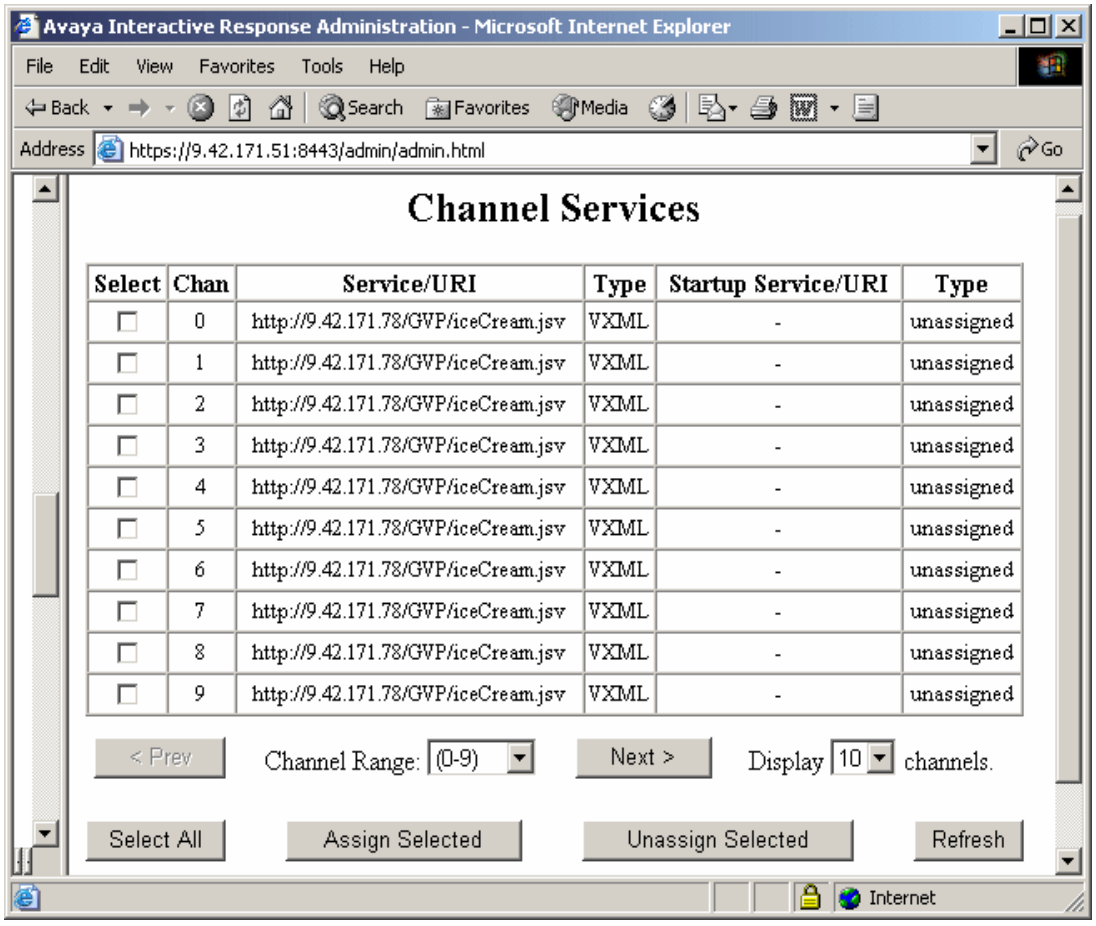
Avaya Interactive Response Administration - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address https://9.42.171.51:8443/admin/admin.html
Go
Channel Services
Select Chan Service/URI Type Startup Service/URI Type
0 http://9.42.171.78/GVP/iceCream.jsv VXML - unassigned
1 http://9.42.171.78/GVP/iceCream.jsv VXML - unassigned
2 http://9.42.171.78/GVP/iceCream.jsv VXML - unassigned
3 http://9.42.171.78/GVP/iceCream.jsv VXML - unassigned
4 http://9.42.171.78/GVP/iceCream.jsv VXML - unassigned
5 http://9.42.171.78/GVP/iceCream.jsv VXML - unassigned
6 http://9.42.171.78/GVP/iceCream.jsv VXML - unassigned
7 http://9.42.171.78/GVP/iceCream.jsv VXML - unassigned
8 http://9.42.171.78/GVP/iceCream.jsv VXML - unassigned
9 http://9.42.171.78/GVP/iceCream.jsv VXML - unassigned
< Prev
Channel Range: (0-9)
Next >
Display 10 channels.
Select All
Assign Selected
Unassign Selected
Refresh
Internet

Figure 2-20 Channel Services

Screen capture reprinted by permission from Microsoft Corporation

1. To assign a URI to one or more channels, select the check box next to those channels, or click **Select All**, then click **Assign Selected**. The next window is Assign Service to Channels, as shown in Figure 2-21 on page 34.
2. In this window, first select **VXML_URI** from the Assign menu list, which will cause the window to alter and only display the appropriate field.
3. Enter the URI of your application in the URI field. The To Channels field will be completed based on your selections on the previous window.
4. After entering your URI, you can click **Verify** to make sure that the Avaya IR can resolve the reference and retrieve the file.

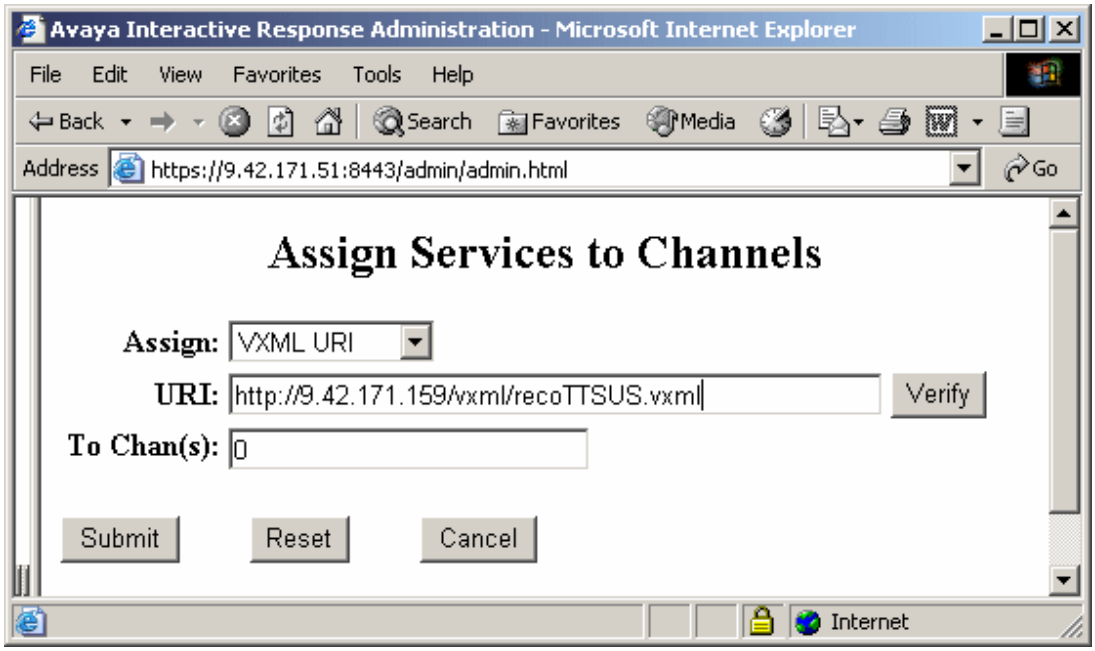

Figure 2-21 Assign Services to Channels
Screen capture reprinted by permission from Microsoft Corporation

Using Number Services

To assign a URI to one or more specific numbers, click **Number Services** in the navigation panel on the left of the Web Administration window. The procedure is similar to the one we described for assigning URIs to channels. You can assign a URI to a single number or a range of numbers, which makes the configuration very simple if the telephony protocol you are using provides number information.

Figure 2-22 on page 35 shows an example of the Assign Number Services window where a URI has been assigned for calls where the number is in the range 5551200 to 5551250. Your are required to enter a value in the first box of the Called Numbers and Calling Numbers fields. In Figure 2-22 on page 35, we have entered the value “any” to allow all calls with called numbers in the specified range to use the application.

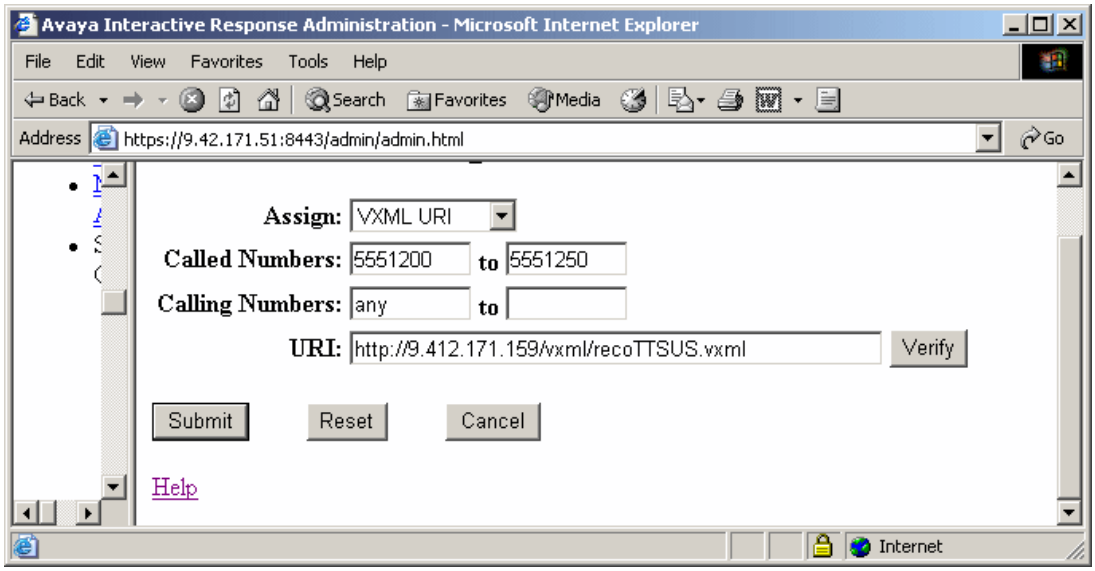

Figure 2-22 Assign Number Services window
Screen capture reprinted by permission from Microsoft Corporation

MRCP configuration file

When you install the MRCP package on the Avaya IR, the /vs/sproxy/cfg/mrcp.cfg configuration file is created. This file sets the values of various parameters related to the internal operation of the MRCP client. We decide that the four parameters described in Table 2-3 are the ones that are most relevant to WebSphere Voice Server. For further information about these parameters and others, refer to the file itself and the manual *Avaya Interactive Response Release 1.3 MRCP Feature*.

Table 2-3 Important parameters in the Avaya MRCP configuration file

Parameter	Description	Default value
PerPortConnection	Indicates whether the MRCP server uses port 554 for multiple per port connections for signaling (TRUE), or a single connection for signalling (FALSE). For WebSphere Voice Server, this parameter should be set to TRUE.	TRUE
BuiltinLocale	The locale of the language to be used for built-in ASR grammars, such as date, digits and currency. You need to change this parameter if you will be using a language other than US English.	en-US
ExtraASRAlloc	This is the number of additional ASR ports to allocate, expressed as a percentage of the administered ports, that is, the ports defined in Speech Recognition Configuration. For example, if the number of administered ports is 48, and ExtraASRAlloc is set to 10, then an additional five ports (10% rounded up) would be allocated. On a busy system, because of the way WebSphere Voice Server manages allocations, ports may not be released quickly enough to keep up with demand. This parameter increases the number of ports available to Avaya IR to increase the probability of ports being available.	0

Parameter	Description	Default value
ExtraTTSAlloc	This is the number of additional TTS ports to allocate, expressed as a percentage of the administered ports, that is, the ports defined in Text-to-Speech Configuration. This parameter is the TTS equivalent of the ExtraASRAlloc parameter.	0

Note: The default values of ExtraASRAlloc and ExtraTTSAlloc are 0. During all load testing, no errors occurred when using 0.

If you find that you experience session allocation failures on Avaya IR when your system is busy, but you have free engines on your WebSphere Voice Server systems and the number of defined ports matches the number of started engines, you should consider setting these parameters to non-zero values.

VoiceXML configuration file

When you install the AVvoicxml2-0 or AVvoicxml package on the Avaya IR, the /vs/data/vxml/defaults.xml configuration file is created. This file sets the values of various parameters related to the internal operation of the VoiceXML Browser. Table 2-4 shows the relevant parameters and the default values assigned to them.

Table 2-4 Avaya IR VoiceXML browser configuration

Browser parameters	Default value configured
promptgender	female
confidencelevel	0.5
sensitivity	0.5
speedvsaccuracy	0.5
bargein	true
caching	fast
audiofetchhint	prefect
documentfetchint	safe
inputmode	dtmf voice

This file also contains the default VoiceXML application that the Avaya IR uses, when the IR has trouble accessing or parsing the assigned VoiceXML application.

Note: We did not change any of the parameters described in Table 2-4 on page 36 for the Interoperability test. For further information about these parameters and others, refer to the file itself and the document *Avaya Interactive Response Release V1.3*.

2.4.4 Third-party packages for Avaya IR

In this section, we discuss third-party packages for Avaya IR.

Real Soft OAMP Tools For Avaya IR

Real Soft Inc provides tools to administer, monitor, and manage the Avaya IR and the voice applications. Furthermore, Real Soft tools offer reporting and analytics that offer application centric reporting as well as platform centric reporting. This section gives a brief overview of the tools. These tools have been certified by Avaya for General Availability. For a more detailed explanation, please go to the following URL:

http://www.realsoftinc.com

The tools available for the Avaya IR are discussed in the following sections.

SNMP Netwatch Monitor 4.2

This is a SNMP based fault management and monitoring tool for Avaya IRs and ScanSoft Speech Servers. Netwatch SNMP Monitor comes with a built-in Castle Rock SNMP Manager, but can also integrate with other SNMP managers, such as IBM Tivoli®. By being able to remotely monitor multiple Avaya IRs and Speech Servers and alert users on traps via e-mail, SMS, and pagers, Netwatch SNMP Monitor provides a robust fault management system.

OneManage 5.1

OneManage simplifies application deployment and management across multiple Avaya IR systems and Application Servers from a centralized location. In such a distributed environment, OneManage enables administrators perform the following tasks:

- Application deployment and distribution
- Application management and version control
- Management of a large distributed Application Development Environment
- Assignment of applications to the channels
- Reports on deployment activity, call detail, and call summary

Key features of OneManage:

- One Step wizard enables deployment of DTMF and Speech applications for Avaya IVR systems located across multiple locations. It also enables deployment of J2EE based VoiceXML applications where the speech application resides on an application server or Web server.
- Application Definition and Packaging enables application deployment with channel assignment and integration with source control management. The application to be managed by OneManage can be a speech application, prompt library, grammar definitions, or system configuration files.
- Performs pre- and post-deployment validation checks to ensure the application is deployed on the right platform.
- Provides the ability to rollback to older versions of a given application.

Admin Central 1.0

Admin Central, as a product suite, enables administrators and production staff to administer and monitor speech applications and the speech platform to ensure service quality.

Admin Central is comprised of a thin client Graphical User Interface (GUI), as shown in Figure 2-23, that enables administrators to remotely administer and monitor one or many Interactive Voice Response (IVR) systems and Speech Applications. The GUI also enables users to request and view custom and aggregated reports.

Admin Central's monitoring capability comes from the built-in SNMP agent for the Avaya IRs. This information can be viewed from any SNMP Manager as well as the Admin Central client. The Admin Central client also provides a way to change a thresholds for the monitored parameters.

Tip: The Admin Central client is very convenient when a client needs a centralized administration and management console to manage an IR cluster.

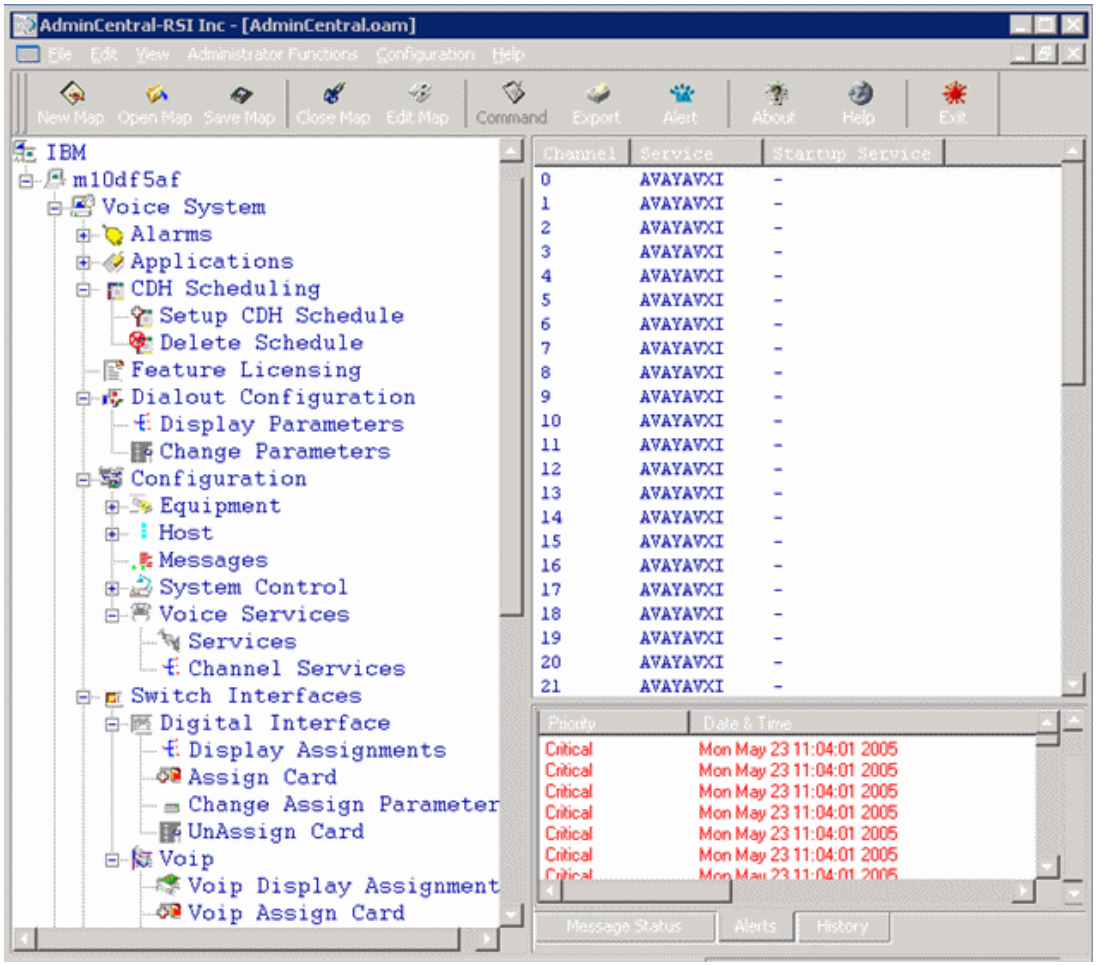

Figure 2-23 Admin Central Client window

In terms of measuring the efficiency of the systems, Admin Central lets users establish thresholds for metrics, including availability and mean time between failures. This allows users to track metrics for individual systems or for group of systems.

Admin Central comes bundled with a notification server called Insta Alert (see "Insta Alert 1.0" on page 39) that reports and routes alert through multiple channels and takes user defined actions like Failover Recovery after detecting failures.

Insta Alert 1.0

Insta-Alert Notification Engine provides a SDK that enterprise applications can integrate to, providing notifications and alerts.

Messages can be posted to the Insta Alert via a HTTP Post or by using a JDBC adapter or file adapter that pulls data from an outside data source.

The Rules Engine in Insta Alert allows users to establish thresholds, and allows users to receive alerts based not on individual alerts, but instead on specific combination of metrics for certain systems.

Figure 2-24 shows an example of creating a new service definition with the Insta Alert user interface.

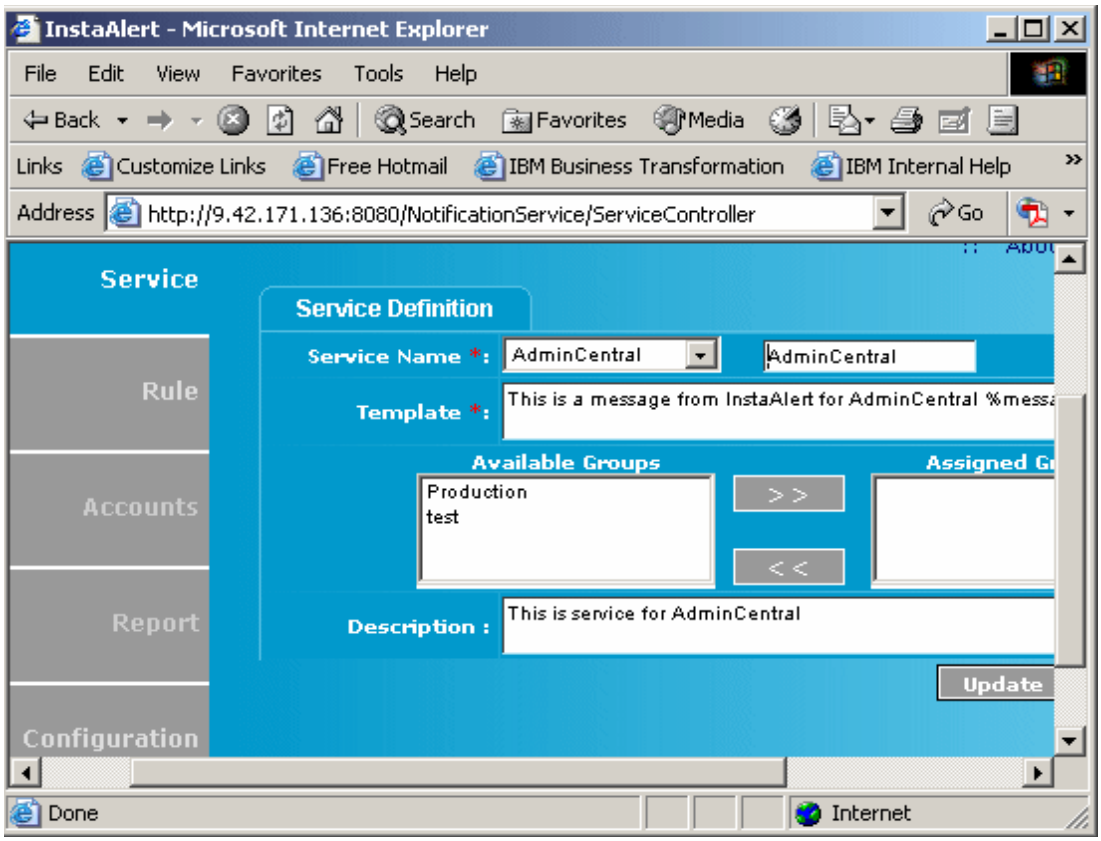

Figure 2-24 Insta Alert User Interface

Screen capture reprinted by permission from Microsoft Corporation

They key benefits are:

- Provides notification/alerting functionality to existing applications like SNMP Netwatch
- Event based and time based notifications
- Multicasting, broadcasting, and alternate routing of notifications
- Notification of alerts through multiple communication channels, such as e-mail, SMS, phone, and pager. A Web-based archival system can also be published for all notifications

Cleo TN3270/5250 for Data Integration

Cleo's TN3270/5250 is designed for connectivity to high-performance mainframe servers. For a detailed description of the products and services, go to the following URL:

http://www.cleo.com

Gold Systems Vonetix for Data Integration

Vonetix provides a single infrastructure for integrating IVR applications and speech recognition technologies with an enterprise's existing client communication interfaces. It offers a portfolio of plug-ins to further enhance connectivity to enterprise applications and Web-based applications. For more information, go to the following URL:

http://www.goldsystems.com

2.5 Development environment for Avaya IR

This section explores the Avaya Speech Application Builder (Avaya SAB) platform as well as the IBM Voice Toolkit for WebSphere Studio as two options to write voice applications for the Avaya IR.

2.5.1 Avaya Speech Application Builder development prerequisites and planning

Avaya IR can be complemented by the Avaya Speech Application Builder (Avaya SAB) platform, which is essentially a network based tool that allows development teams and call centers to build interactive voice response applications. It is targeted to run against Avaya IR and fully integrates to the WebSphere Application Server and WebSphere Voice Server V5.1 (see Figure 2-25 on page 41).

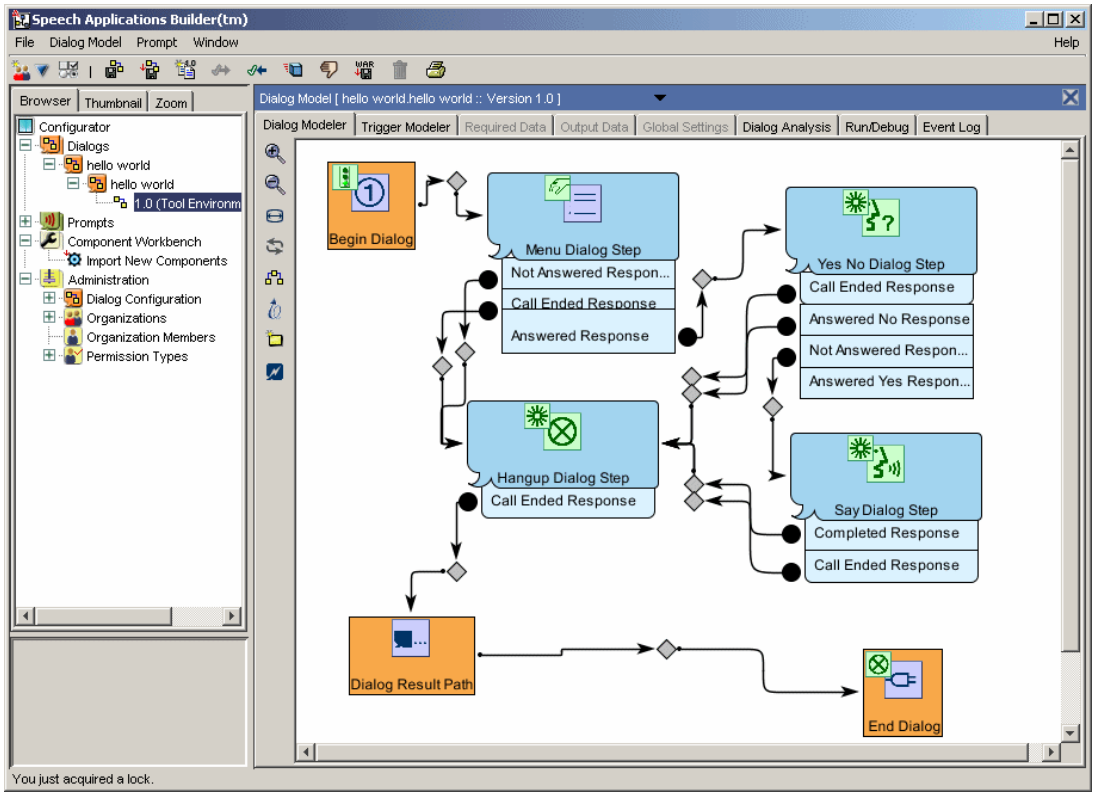

Figure 2-25 Configurator interface in Avaya SAB

The two main components of the Avaya SAB are the Configurator, a graphical user interface used to design the applications, and the Avaya SAB runtime environment, which executes them. The main strength behind the Avaya SAB is the library of ready-made components, which allow the user to quickly create or reuse typical dialog components as objects in a visual interface. Please refer to Figure 2-26 on page 42 for an example of a component edit window.

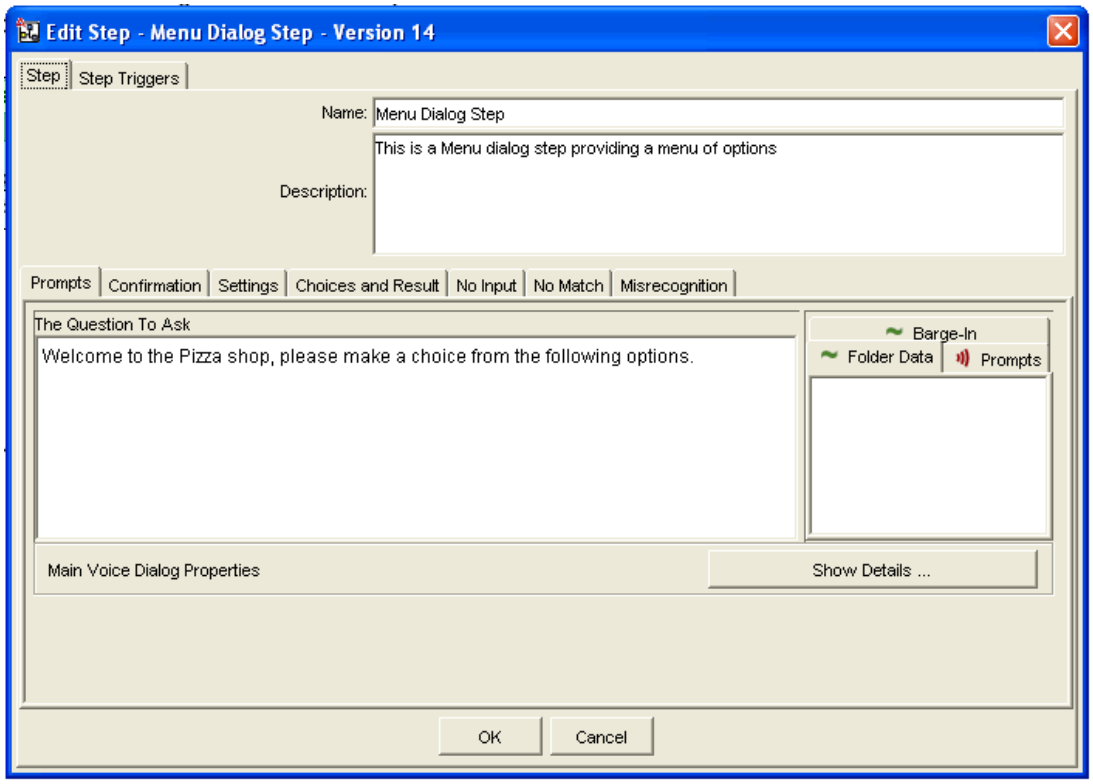

Figure 2-26 Avaya SAB component detail

Avaya SAB is a J2EE-based environment and application, independent of the underlying voice and telephony platform it is designed to operate on, and therefore is deployable in a wide variety of environments.

Avaya SAB V1.3 is the version that we used in the lab test for this Redpaper. The minimum requirements to be able to successfully run the environment are:

- Voice Server platform
  - Avaya IR Release V1.3 with WebSphere Voice Server V5.1.2
- Application platform
  - Avaya IR Release V1.3 with WebSphere Application Server V5.1.1
- Configuration databases
  - Firebird (installed as the default database for Avaya SAB)
  - Microsoft SQL Server 2000
- Test environment system requirement
  - 512 MB of RAM
  - 1 GHz Processor
  - Microsoft Windows 2000 Professional

System requirements

When installing the Avaya Speech Application Builder, there are some considerations to keep in mind.

It is a best practice to keep the configuration and reporting databases on physically separate servers because the configuration database, unlike the reporting database, is typically dropped during deployment.

The native database management system included with the Avaya SAB is the Firebird RDBMS. It is not uncommon practice to use the Microsoft SQL Server as an alternate RDBMS due to its comprehensive Enterprise Manager environment.

Figure 2-27 is a diagram demonstrating a typical Avaya SAB environment deployment.

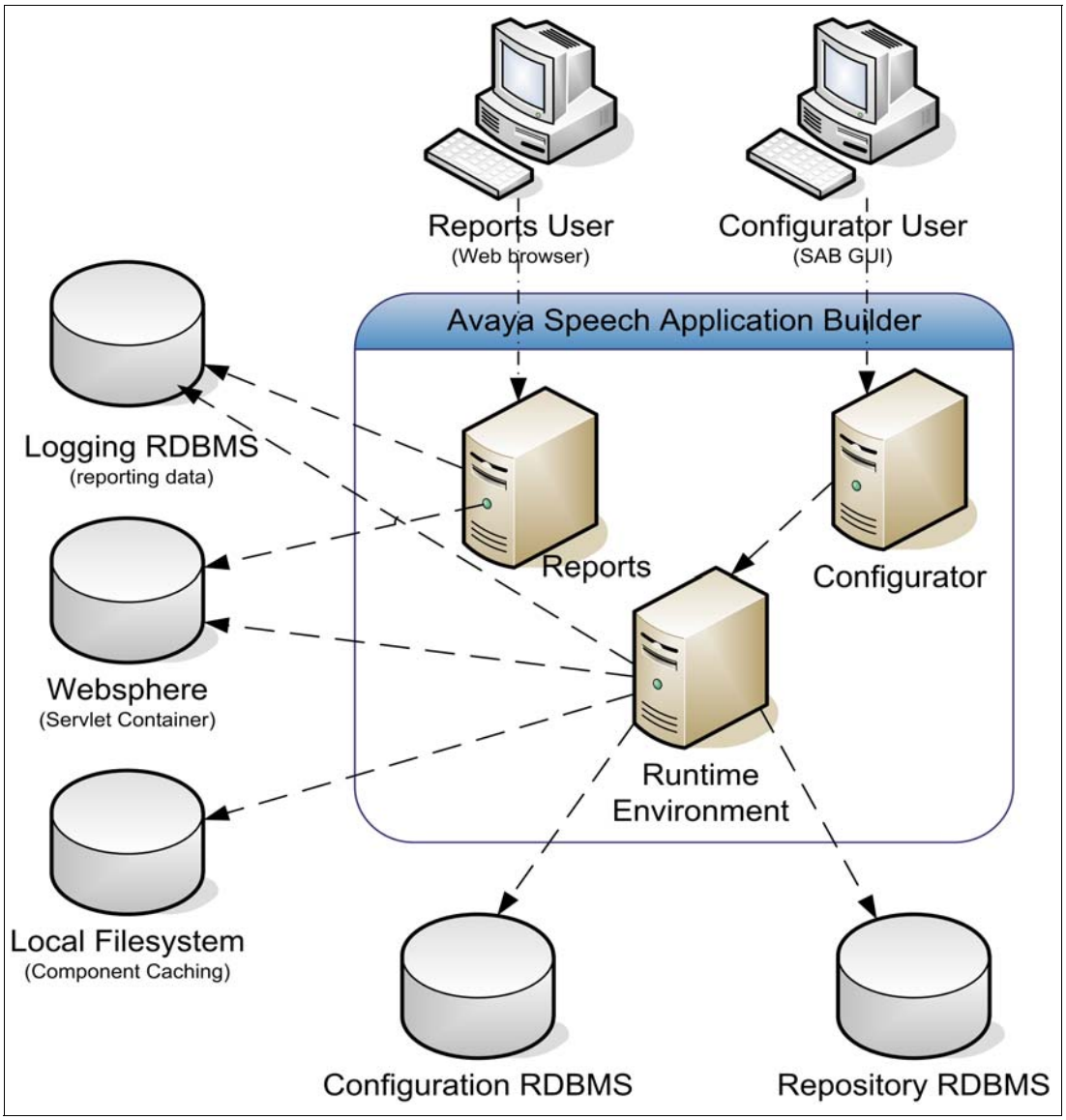

Figure 2-27 Avaya SAB dependency diagram

The different database stores are explained below.

- Logging RDBMS: An SQL database to which the Runtime subsystem logs data for extraction by the Reports subsystem.
- Servlet Container: A J2EE compliant servlet container used by both the Report and Runtime subsystems. Supported containers include JBOSS, JRUN and WebSphere.
- Local Filesystem: The file system on the node containing the ServletContainer in which the Runtime subsystem is running. The Local Filesystem is used to cache versions of component JARs.
- Configuration RDBMS: An SQL database used to store configuration data for the Runtime subsystem.
- Repository RDBMS: An SQL database used by the Runtime and Configurator subsystems to store versioned dialog flows and components.

Software configuration

The Avaya SAB tool consists of four separate media CDs.

1. Avaya Runtime Media
2. Avaya Core Development Media
3. Avaya Intermediate Development Media
4. Avaya ICM Media

For more in-depth information about Avaya SAB, refer to the documentation (in PDF format) that describes the development tool at:

`http://www.avayadocs.com`

2.5.2 IBM Voice Toolkit for WebSphere Studio

The IBM Voice Toolkit V5.1 for WebSphere Studio is an enhanced collection of technologies based on Eclipse technology that make it simple for you to create sophisticated and successful VoiceXML applications without having to know the internals of voice technology. The Voice Toolkit for WebSphere Studio provides the tools to build, debug, and deploy voice applications across the IBM WebSphere family. When it is installed, it is an integral part of WebSphere Studio, providing a single workbench for the creation of all aspects of a business application. The workbench provides facilities to create Java components, servlets, JSPs, EJBs, Web pages, voice components, and the ability to test and debug all these components individually or together on the desktop.

The Voice Toolkit for WebSphere Studio features a full-featured voice development environment, including graphical call flow building, VoiceXML development and debugging, grammar development and testing, pronunciation builder, and Call Control eXtensible Markup Language (CCXML) development.

Refer to 3.3, “A Simple VoiceXML Application using Voice Toolkit for WebSphere Studio and Call Flow Builder” on page 49 for usage information.

Hardware and software prerequisites

The hardware requirements for the Voice Toolkit for WebSphere Studio are:

- A system suitable for running Microsoft Windows 2000 Professional or Microsoft Windows XP Professional
- At minimum, an Intel Pentium® 500 MHz processor or equivalent
- 768 MB of RAM
- 150 MB of disk space
- 150 MB of temporary space on the on the drive specified in your user TMP environment variable
- A display adapter setting of at least 256 colors with a minimum resolution of 800x600 (1024x768 or higher is recommended.)

The software requirements are:

- Microsoft Windows 2000 Professional (Service Pack 4 or higher) or Microsoft Windows XP Professional (Service Pack 1 or higher)
- Adobe Acrobat Reader V4.0.5 or later
- An installed version of one of the following: WebSphere Studio Site Developer V5.1.2, WebSphere Studio Application Developer V5.1.2, or WebSphere Application Server-Express V5.1.2.

The MRCP grammar test tool requires network access to a WebSphere Voice Server V5.1 system.

Installation and configuration

The Voice Toolkit for WebSphere Studio is packaged with the WebSphere Voice Server V5.1 product, as is WebSphere Studio Site Developer V5.1.2. I can also be downloaded from:

`http://www.ibm.com/software/pervasive/voice_toolkit`

The Voice Toolkit for WebSphere Studio includes an ASR engine and a TTS engine. During the installation process, the language files for these engines are downloaded from the Internet, so you need to make sure your system can access the following URL:

`http://www3.software.ibm.com/ibmdl/pub/software/websphere/voicetoolkit`

We recommend that you use a high-speed connection to the Internet and only install one language at a time because the language files vary in size from 300 Mb to 700 MB. The Voice Toolkit for WebSphere Studio does support multiple languages, so if you need to a second or third language, these can be installed after the initial install.

3

Developing voice-enabled applications

This chapter describes several different applications that can be used to design, create, and deploy VoiceXML applications for the Avaya IR V1.3. A simple ASR and TTS application that was used in the testing at the ITSO will be discussed. There is also some discussion of VoiceXML and related specification (for example, SSML) limitations and considerations.

© Copyright IBM Corp. 2005. All rights reserved.

3.1 What is VoiceXML?

The Voice eXtensible Markup Language (VoiceXML) is a World Wide Web Consortium (W3C) standards-based markup language that integrates voice technology with Web-based development offering flexible, powerful, and portable Interactive Voice Response (IVR) applications. Please visit the following URL for more information:

http://www.w3c.org

Since VoiceXML is standards-based, IVR applications may be used across many vendor specific platforms. There are several choices of VoiceXML development environments, from a simple text editor to vendor specific tools, such as IBM Voice Toolkit V5.1 for WebSphere Studio, Avaya Speech Applications Builder (Avaya SAB) V1.3, and Avaya IVR Designer. This chapter discusses Voice Toolkit for WebSphere Studio and Avaya SAB in more detail.

3.1.1 VoiceXML key benefits

VoiceXML has most, if not all, of the capabilities of traditional proprietary IIVR systems.

Note: In migration situations, check the IVR Vendor's documentation, since some proprietary programming environments may have functions that were not extended to the VoiceXML browser implementation.

Because VoiceXML is Web-based and normally runs on Web application servers or Portal Servers, it can be tightly integrated with a variety of Web applications, thereby enabling greater flexibility. Since VoiceXML is standards-based, it is highly portable and does not need to run on proprietary systems, which offers greater flexibility.

3.2 A brief overview of VoiceXML specifications

VoiceXML is a World Wide Web Consortium standard (W3C). Therefore, the specifications of VoiceXML are standardized and listed on the W3C Web site. As with most standards, some portion of the standard are *must* have, *required*, *shall*, while other portions are *should have*, *recommended*, *may have*, and *optional*. Therefore, each vendor's implementation of the standard will vary. This not only applies to the VoiceXML V2.0 standard, but also the related W3C standards, for example, SSML and SRGS. Further information regarding the VoiceXML V2.0 standard can be found at the following URL:

http://www.w3.org/TR/voicexml20

3.2.1 Elements

VoiceXML V2.0 has an array of elements that can go into a VoiceXML document. For a list of elements and their functions in the VoiceXML V2.0 standard, please visit the following URL:

http://www.w3.org/TR/voicexml20

3.2.2 Common attributes

VoiceXML has attributes that can be assigned to a VoiceXML document. An example of common attributes include version and language. For a list of attributes and their definitions for the VoiceXML V2.0 standard, please visit the following URL:

http://www.w3.org/TR/voicexml20

3.3 A Simple VoiceXML Application using Voice Toolkit for WebSphere Studio and Call Flow Builder

There is really nothing special that needed to be done to the following VoiceXML application built on Voice Toolkit Version for WebSphere Studio V5.1, which is a toolkit extension to WebSphere Studio Application Developer Version V5.1.2, to get it to run on the Avaya IR Version 1.3.

Note: There are many options in creating an application like the following to yield similar results.

In the simple sample application called pizza, a caller will be greeted and prompted for one of three types of pizza. The caller will be offered to order either a cheese pizza, one with pepperoni, or an all dressed (everything) pizza. After recognition of the type, the caller will be asked if the would like to order another.

Note: The Help system on the WebSphere Studio Application Developer for Voice Toolkit has several tutorials that may be useful to the inexperienced to create an application.

3.3.1 Creating the Application with Call Flow Builder

Begin by opening the WebSphere Studio Application Development environment and follow the tutorial to create a Call Flow Builder project (see Figure 3-1).

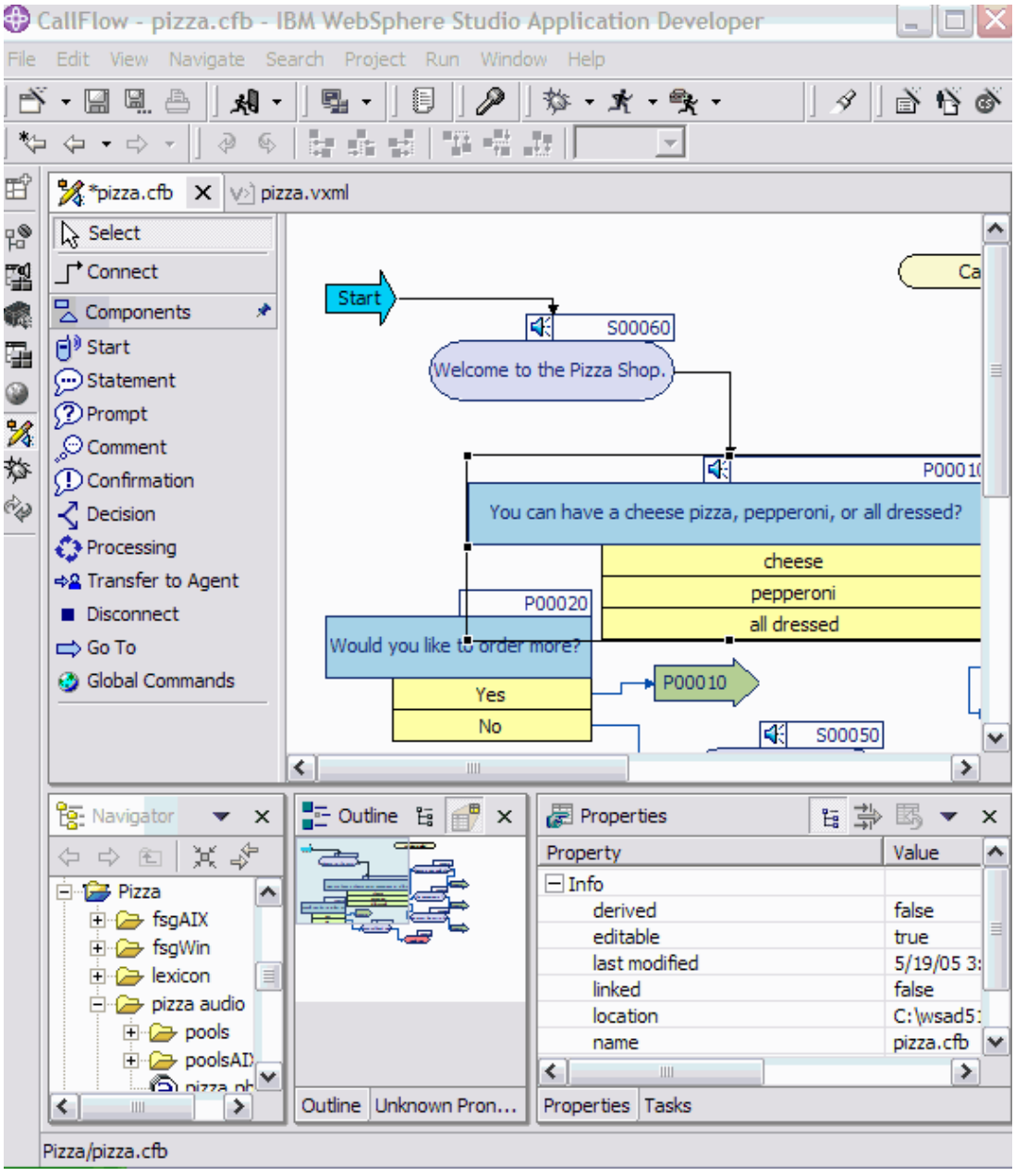

Figure 3-1 Call Flow Builder View

The following steps were used to created the pizza shop sample application in Call Flow Builder:

- Start.
- Statement: “Welcome to the Pizza shop”.

- Prompt: “You can have a cheese pizza, pepperoni, or all dressed. Which would you like?”
  - User Response: “cheese”
  - User Response: “pepperoni”
  - User Response: “all dressed”
- Statement: “cheese is nice and cheesy”.
- Statement: “the pepperoni is fresh”.
- Statement: “you must be extra hungry’.
- Three Go Tos target each of the above statements.
- Prompt: “Would you like to order more?”
  - User Response: “Yes”
  - User Response: “No”
- One Go To for the Yes response.
- Statement: “Please call back soon”.
- Disconnect

Please see Figure 3-2 for the complete Call Flow application.

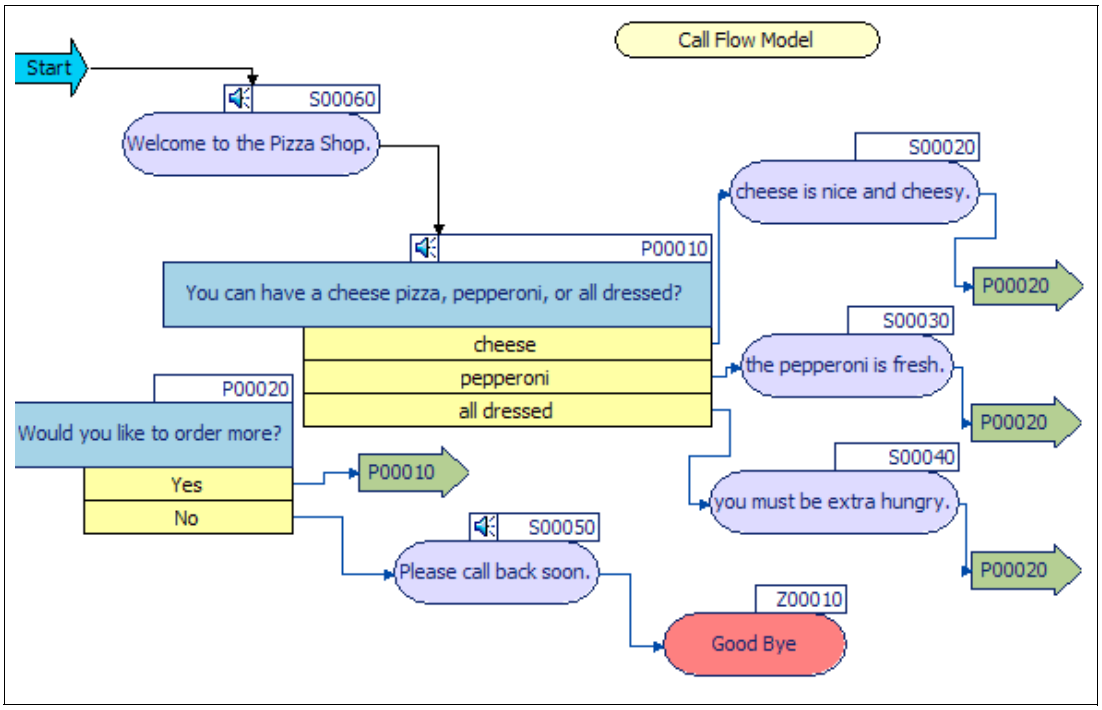

Figure 3-2 Pizza application from Call Flow Builder

One of the features of Call Flow builder is that it allows an audio prompt to be recorded for each *statement* and *prompt*. To record the prompt, use your headset to record the prompt right from the Call Flow builder view. To record the audio prompt, click the audio icon, as shown in Figure 3-3.

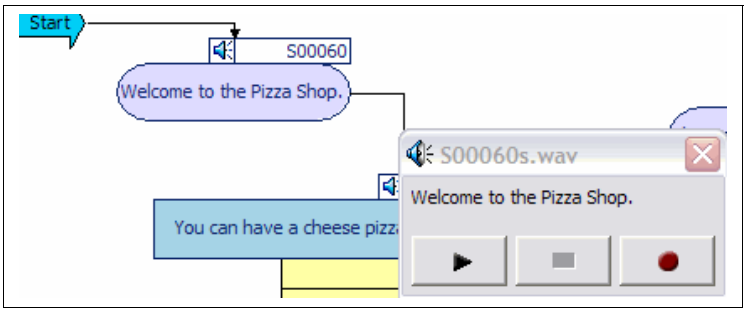

Figure 3-3 Recording the Welcome prompt audio file

When Call Flower builder assumes there will be an audio prompt, an *audio* VoiceXML statement is generated. To force a *prompt* statement, the audio icon, must be removed from each item, as shown in Figure 3-4.

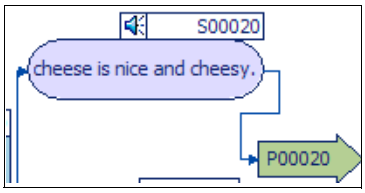

Figure 3-4 Statement before audio icon is removed

To remove the audio icon, select the item, in this case, statement S00020, as shown in Figure 3-5.

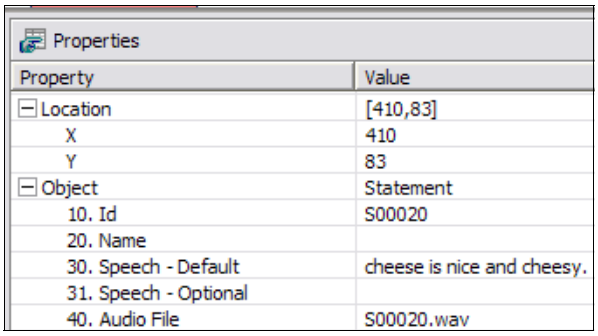

Figure 3-5 Properties box for statement S00020

Clear the S00020.wav from the Properties box, in this case, line 40, as shown in Figure 3-6 on page 53.

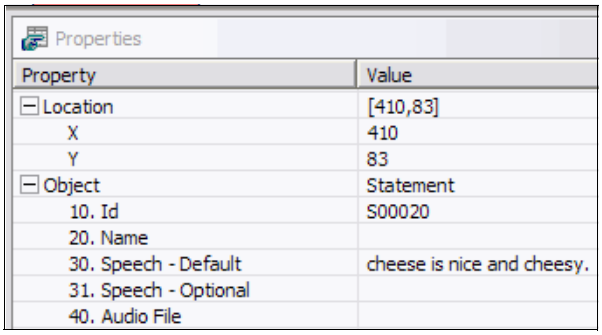

Figure 3-6 Properties with audio file cleared

The graphical window will confirm the deletion by removing the audio icon from the item, as shown in Figure 3-7.

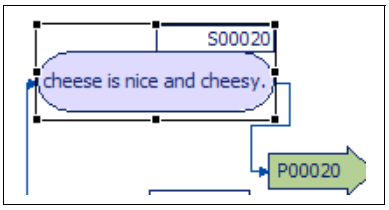

Figure 3-7 Statement S00020 with audio icon removed

When the application is complete, right-click the white space and select **Verify**. The verify operation will either indicate the errors that need to be corrected or generate a “Validation successful” box, as shown in Figure 3-8.

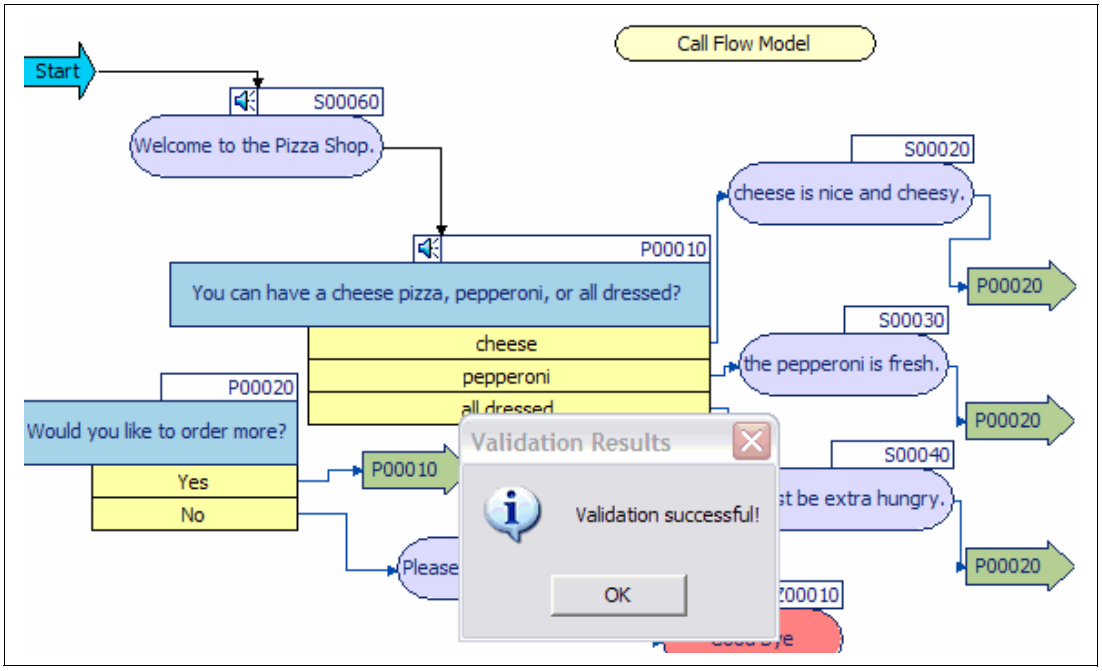

Figure 3-8 Call Flow application has been verified

The Call Flow builder tool allows the application to be tested by simulating both speech and DTMF calls. When the testing is complete, code can be generated by right-clicking the white space. In the test at the ITSO, we chose to test VoiceXML. Before the VoiceXML code is generated, you will be prompted to save (please see Figure 3-9).

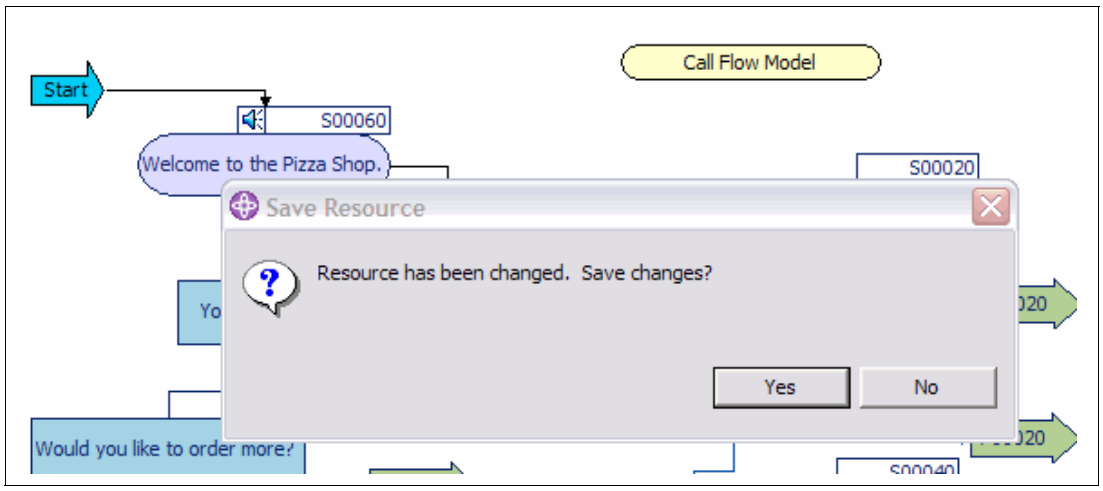

Figure 3-9 Saving before the VoiceXML code is generated

After the save, a window similar to Figure 3-10 will be opened, which contains the VoiceXML code. The full text of the code can be found in Appendix A, “Sample code” on page 87.

pizza.cfb pizza.vxml

```
1 <?xml version="1.0" encoding="iso-8859-1"?>
2
3 <!DOCTYPE vxml PUBLIC "-//W3C//DTD VOICEXML 2.0//EN" "vxml20-1115.dtd">
4 <vxml xmlns="http://www.w3.org/2001/vxml" version="2.0" xml:lang="en-US"
5
6     <!--Call Flow Model-->
7
8     <!--Begin Call Flow-->
9     <property name="universals" value="help cancel" />
0     <property name="audiomaxage" value="1" />
1
2     <var name="mode" expr="'speech'" />
3     <var name="previousResult" />
4
5     <form id="S00060">
6         <block>
7             <audio src="./pizza%20audio/S00060s.wav">Welcome to the Piz:
8             <goto next="#P00010" />
9         </block>
0     </form>
```

Figure 3-10 VoiceXML code generated from pizza Call Flow object

3.3.2 Testing the VoiceXML pizza application

The pizza.vxml application should be tested in Voice Toolkit for WebSphere Studio. Please refer to the Voice Toolkit for WebSphere Studio help and documentation for details on the testing facilities available. Also, please refer to *WebSphere Voice Server for Multiplatforms V5.1.1/V5.1.2 Handbook,* SG-6447.

Once the testing is complete, the files can be moved to an application server that can be reached from the Avaya IR.

Deploying the application on the Web server

The pizza.vxml file plus the generated WAV files were exported as a file system using the WebSphere Studio export feature. Then the files were moved to an IBM HTTP Server and placed in the *root*\WebSphere\IBMHTTPServer\htdocs\en_US directory.

Access to the pizza.vxml can be tested to verify the application server is functioning by using a visual browser and point it toward the VoiceXML file.

Note: It was found that Microsoft Internet Explorer V6.0 SP1 had a problem displaying the actual VoiceXML file due to the following statement:

```
<!DOCTYPE vxml PUBLIC "-//W3C//DTD VOICEXML 2.0//EN" "vxml20-1115.dtd">
```

One solution would be to comment out the above line in the VoiceXML file or to use a different browser.

Please refer to 2.4.3, “Avaya IR V1.3 configuration” on page 20 to configure the Avaya IR.

3.4 Creating a simple VoiceXML application using Avaya SAB V1.3

As an example, an application was created similar to the one described in 3.3, “A Simple VoiceXML Application using Voice Toolkit for WebSphere Studio and Call Flow Builder” on page 49.

The application is a basic pizza shop ordering system that prompts the user to select between three choices. Since the Avaya SAB Configurator is purely a visual component environment requiring drag and drop action to build the application, setting up a basic dialog flow was relatively simple.

Note: Access to VoiceXML code for direct manipulation is not possible via the configurator and different from the WebSphere Voice Toolkit in that regards. The configurator’s ease of use stems from the reality that even non technical personnel can easily manipulate the call flow without prior knowledge of VoiceXML programming granted they have a basic level of training on SAB.

There is somewhat of a learning curve that can be easily overcome if you have the proper user guides. These are all documented in PDF format and can easily be downloaded, as described later in this chapter.

3.4.1 Call flow overview

The call flow of our Avaya SAB V1.3 application is detailed below:

- IVR: Welcome to the Pizza Store.
- IVR: You can have a cheese, pepperoni or all dressed pizza. What pizza would you like?
  - Response: Cheese.
    - IVR: Cheese is nice and cheesy.
  - Response: Pepperoni.
    - IVR: Pepperoni has a lot of pepper.
  - Response: All dressed.
    - IVR: You must be extra hungry.
  - ELSE any other request or low confidence level score.
    - IVR: I did not understand.
- IVR: Would you like to order another pizza? Say yes or no.
  - Response: yes.
    - IVR will ask the preceding question and re-run routine.
  - Response: no.
    - IVR: Thank you for your business. Please call back soon.
- Disconnect.

3.4.2 Using the Configurator component

The configurator window, as shown in Figure 3-11 on page 57, is made up of a visual representation of the dialog flow, which is fairly easy for a non-technical person to understand. The code behind it is not accessible for quick manipulation or modification and the user is limited to the property settings that are available on a per component basis.

The components can be retrieved simply by clicking the Add Component menu icon, represented by a folder image, on the thin left hand margin next to the visual dialog modeler.

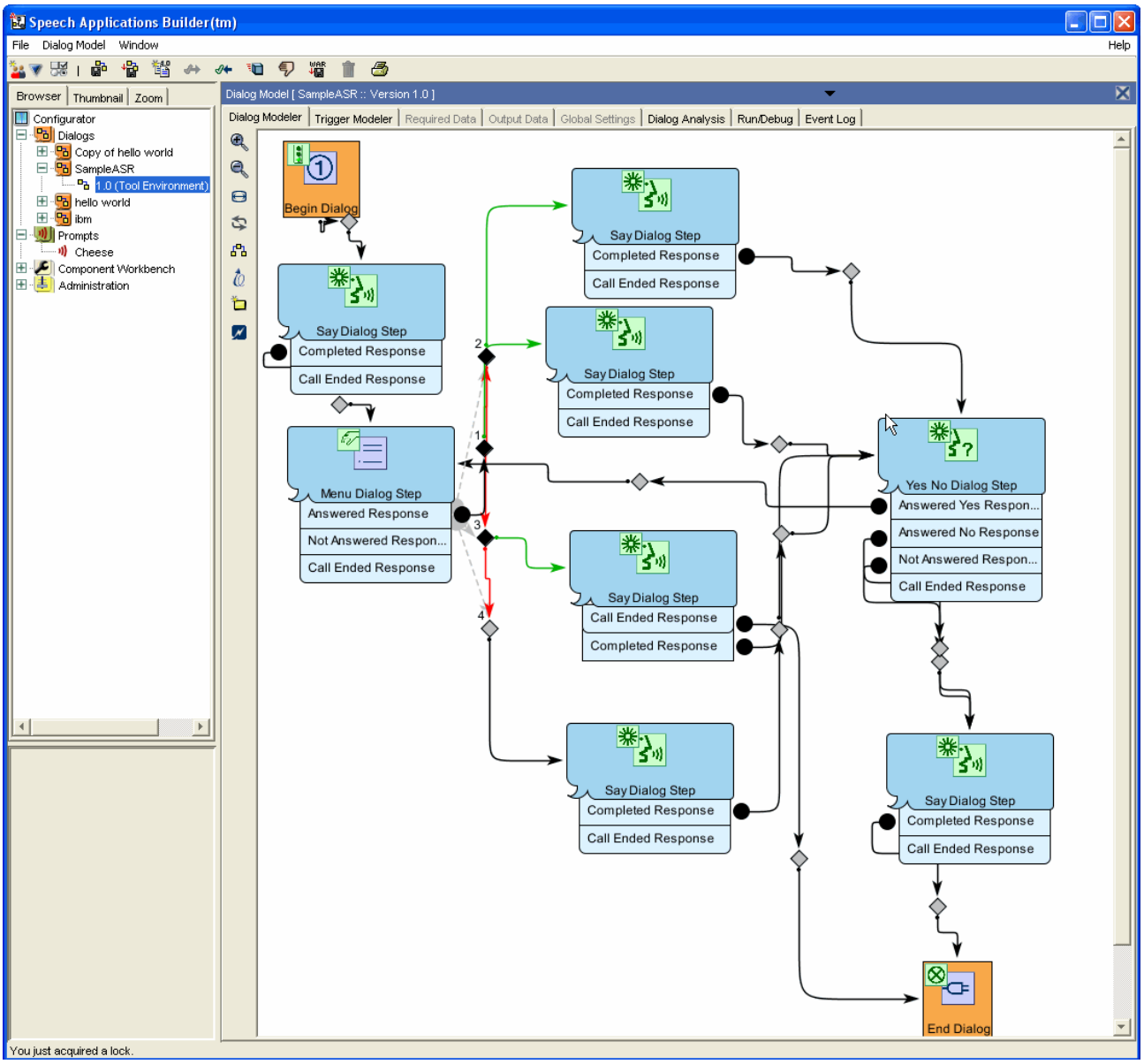

Figure 3-11 Studio Application Builder Configurator component window

The flow of the dialog is actually determined by the connectors used to tie together the different components. By default, each connection contains a decision point that is represented by a grey diamond on the center of the connector. Each decision point module can be modified to contain a conditional statement that in turn affects the dialog flow of the call.

In a typical scenario a menu dialog component would be used. In Figure 3-12, the assigned developer or even a call center manager could alter the text that the system could use to prompt the user. The barge-in feature could also be enabled through this component step by selecting the barge-in option via the tab located on the right-hand side of the window.

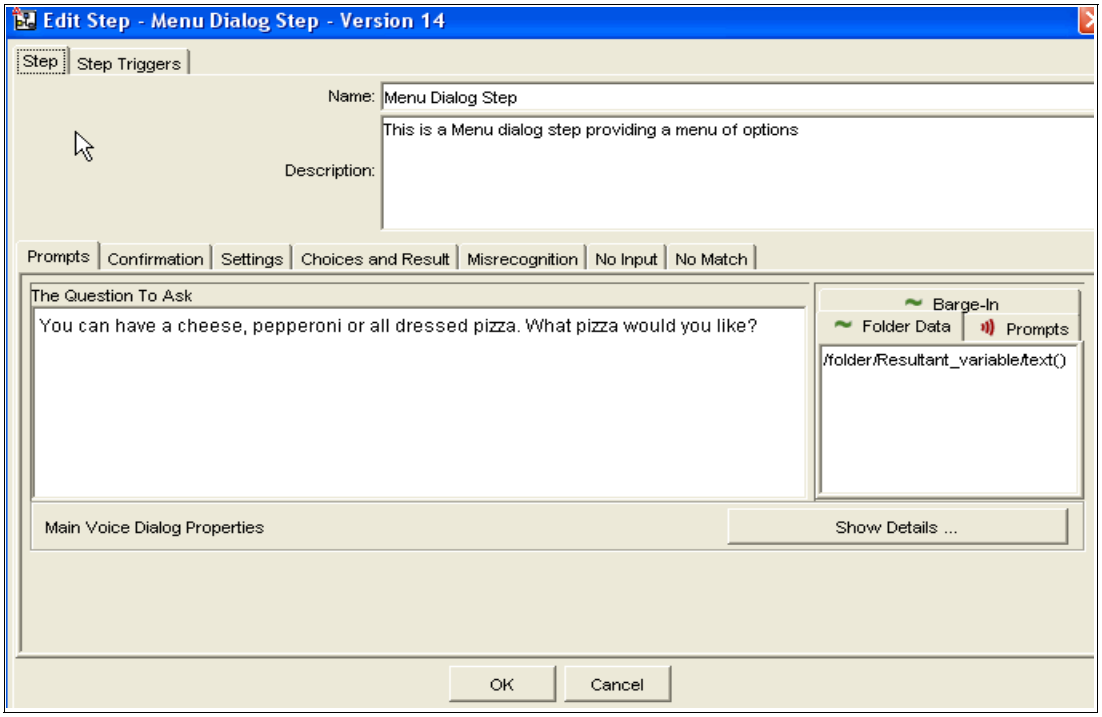

Figure 3-12 Prompt section of dialog component in SAB V.1.3

By clicking **Choices and Results**, the user would list the choices to be given to the caller during the dialog flow, as shown in Figure 3-13 on page 59.

In this example cheese, pepperoni, and all dressed are selected and a resultant variable is created in order to trap the result. This variable can then be used to run against a conditional statement to ensure the proper call flow.

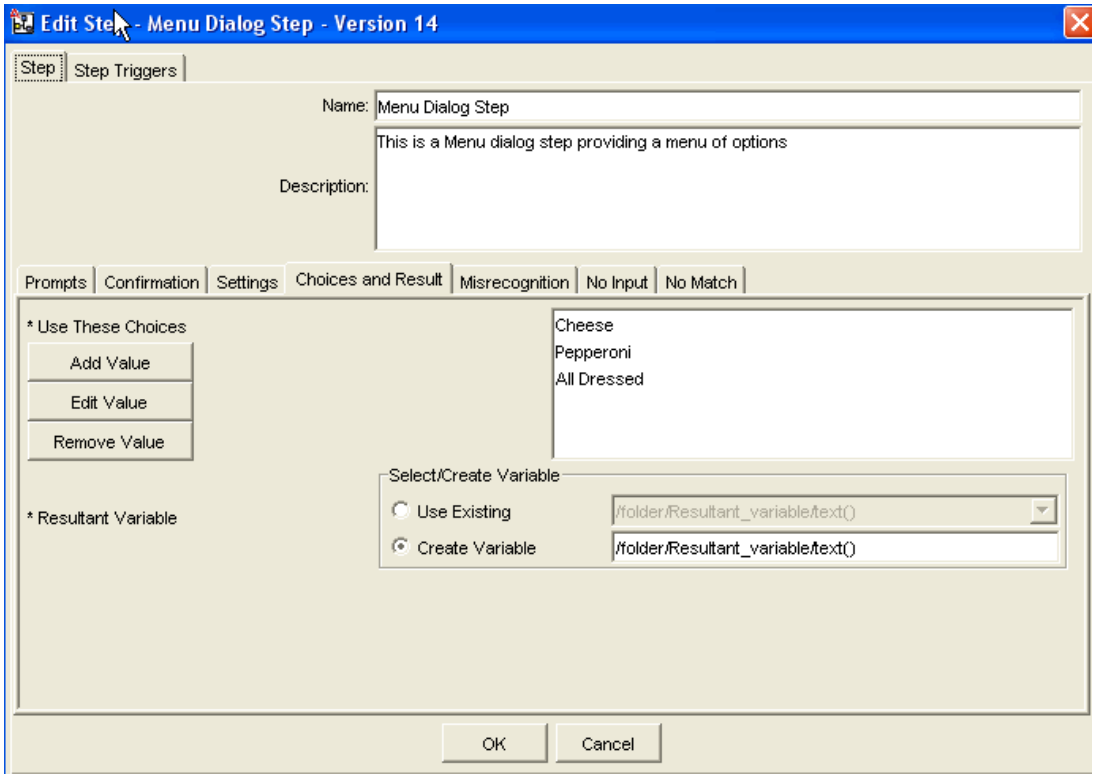

Figure 3-13 Menu selection section of dialog component in Avaya SAB V1.3

The configurator allows you to view the selected properties of the components by listing them on the bottom left-hand corner of the development environment, as shown in Figure 3-14 on page 60.

This is a very high level view of what can be done with one of the many components included with the Avaya SAB V1.3.

Name	Menu Dialog Step
Descrption	This is a Menu dialog step providing ...
Valid?	Yes
Question To Ask	You can have a cheese, pepperoni
Confirmation Question	
Explicit Confirmation	false
Confirmation Acceptance Confidenc...	0.65
Maximum reprompts allowed for co...	1
Menu Options	[Cheese, Pepperoni, All Dressed]
Resultant variable	/folder/Resultant_variable/text()
Prompt for misrecognition event	com.netdecisions.agility.fw.kernel.b...
Maximum disambiguation prompts all...	3
Main Voice Dialog Properties	com.fluencyvoice.runner.dialog.voic...
No Input Voice Dialog Properties	com.fluencyvoice.runner.dialog.voic...
No Match Voice Dialog Properties	com.fluencyvoice.runner.dialog.voic...
Misrecognition Voice Dialog Properti...	com.fluencyvoice.runner.dialog.voic...
Confirmation Voice Dialog Properties	com.fluencyvoice.runner.dialog.voic...
Confirmation No Input	
Confirmation No Match	
Maximum reprompts allowed	3
Acceptance Confidence Threshold	0.65
Acceptance Confidence Margin	0.01
Prompt for no match event	com.netdecisions.agility.fw.kernel.b...
Prompt for no input event	com.netdecisions.agility.fw.kernel.b...

Figure 3-14 Detail of the property view section

All of the dialog components that perform ASR in Avaya SAB V1.3 also handle the following:

- Confirmation: “Did you say 123456?”, if required
- No input: “I’m sorry. I did not hear you, please say it again”
- No match: “I’m sorry I did not understand, please say it again”
- Misrecognition

For more in-depth detail about the usage of this application, please refer to the Avaya documentation provided with the installation CD or search with the keyword “SAB V1.3” at Avaya’s documentation Web site:

http://www.avayadocs.com

Note: The Avaya SAB V1.3 environment does not allow you to access VoiceXML code in order to allow direct manipulation. The VoiceXML is generated dynamically (on-the-fly) in the runtime environment as the call flow is called.

For demonstration purposes, XML code from the Configurator environment was extracted via the VRA files generated by the Avaya SAB for export purposes. The code is entirely generated in XML Version 1.0 and can be viewed in Appendix A, “Sample code” on page 87. It clearly demonstrates the Java class references within the application.

The exposed code is a simple “hello world” application as the volume for the pizza shop application was too large and the “hello world” application demonstrated enough code to give the more experienced IT specialist enough insight to understand the inner workings of the Avaya SAB tool.

3.4.3 Configure and deploy the Application on WebSphere Application Server

Speech Applications developed using Avaya SAB V1.3 are packaged for deployment as a standard J2EE Web Archive or WAR file. This archive groups together all of the necessary code and configuration files according to a simple structure understood by J2EE application servers. In order to deploy the WAR file on an Application Server and assign the channels of the IR to invoke the speech application that is bundled in the WAR file, we need to go through a series of steps.

Step 1: Generate the WAR file

For a complete description of the contents of the Avaya SAB's WAR file, please refer to *Avaya Speech Application Builder's Configuration and Deployment Guide*. This document can be found on the Avaya SAB V1.3 installation CD or you can visit the Avaya documentation Web site at:

`http://www.avayadocs.com`

Once the application has been developed using the Avaya SAB V1.3 Configurator, the next step is to create a WAR file. Avaya SAB V1.3 has a WAR file Generation Wizard, as shown in Figure 3-15, which walks the developer through the steps required in creating a WAR file. The WAR File Generation Wizard automates the following:

- Generating an appropriate web.xml file
- Copying configuration files and other additional resources
- Copying code and grammar archives
- Packaging as a single WAR file

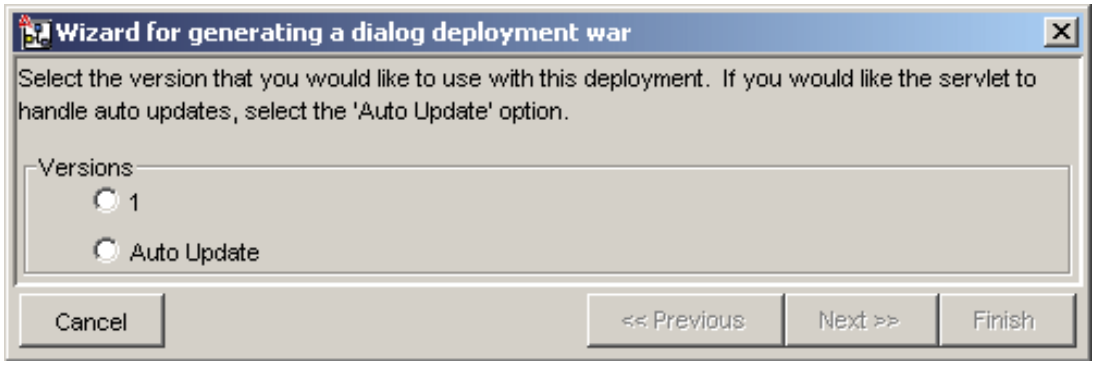
Wizard for generating a dialog deployment war
Select the version that you would like to use with this deployment. If you would like the servlet to handle auto updates, select the 'Auto Update' option.
Versions
1
Auto Update
Cancel
<< Previous
Next >>
Finish

Figure 3-15 Creating a WAR file using the SAB WAR Wizard

The configuration files, other resources, code, and grammar archives included in the WAR are copied directly from the appropriate folders in the Avaya SAB V1.3 Configurator installation, although some unnecessary files are excluded. The WAR does not contain the flows, prompts, or components to be run, or the repository database, which contains them. However, it does need to know which flow to use, which runtime parameters to use for the flow, and how to access the repository database.

Note: Pay attention to the Servlet URI Mapping and Application Server Definition fields in the Avaya SAB wizard for creating a WAR file

Servlet URI mapping

Defines the Servlet URI mapping, and this is the URI the Avaya IR would be using for accessing the speech applications. Application servers typically append the Servlet URI to the WAR file from which the Avaya SAB V1.3 runtime was loaded, without the ".war" suffix. This means that the default value of /servlet/MainDialogFlow will typically make Avaya SAB V1.3 available at:

`http://machine:port/asab/servlet/MainDialog`

if the Avaya SAB V1.3 WAR is called asab.war and machine and port are the address and port respectively on which the servlet container is listening for HTTP requests.

Application server

This selects which application server the WAR file is intended to be used with, in order to accommodate requirements to include or exclude particular system libraries. Make sure that IBM WebSphere is selected during the process of creating the WAR file by the WAR File Creation Wizard.

Step 2: Install WAR File on WebSphere Application Server

The WAR file created by the Avaya SAB V1.3 now needs to be copied over to the machine that is running WebSphere Application Server. The folder location for the WAR File is C:\Program Files\WebSphere\AppServer\installedApps*ServerName*, where ServerName is the host name of the server running WebSphere Administrative Console.

Once this is done, the next step is to "install" the WAR file on the WebSphere Application Server. This process is started by getting on the WebSphere Administrative Console, followed by clicking **Applications** and **Install New Applications**. A screen capture is provided in Figure 3-16 on page 63 to help developers and administrators jump start the process.

Detailed instructions are provided in the *Avaya SAB V1.3 Configuration and Deployment Guide*.

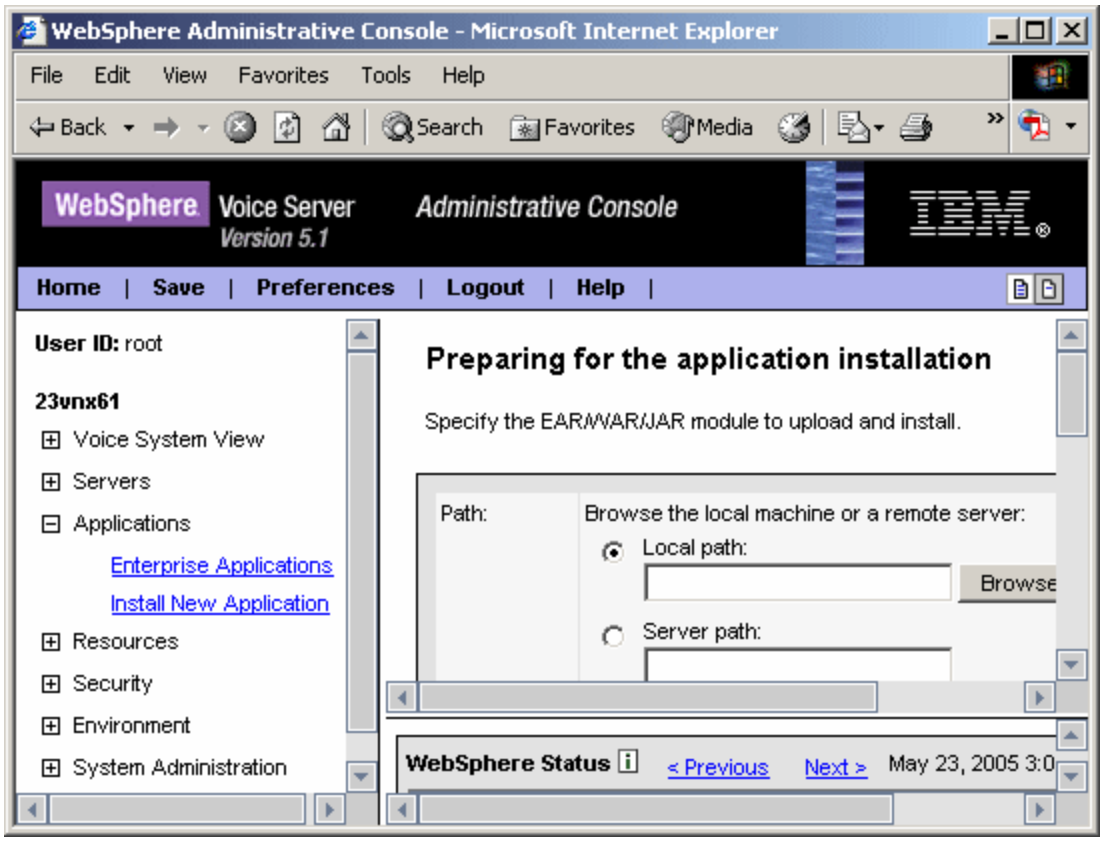

Figure 3-16 WebSphere Administration Console for Installing New Applications
Screen capture reprinted by permission from Microsoft Corporation

This process could be automated by using the Real Soft Inc. OneManage tool that is available for the Avaya IR.

Step 3: Start the newly installed Application on the Application Server

Once the application (WAR file) has been installed, the next step is to start the newly installed Application on the WebSphere Application Server. When you click **Enterprise Applications**, it shows a list of Applications currently installed. Select the speech application installed and click on Install, as shown in Figure 3-17.

Note: There cannot be two Avaya SAB V1.3 applications running at the same time, so the second one will not run, as there can be only be one instance of the Avaya SAB V1.3 runtime.

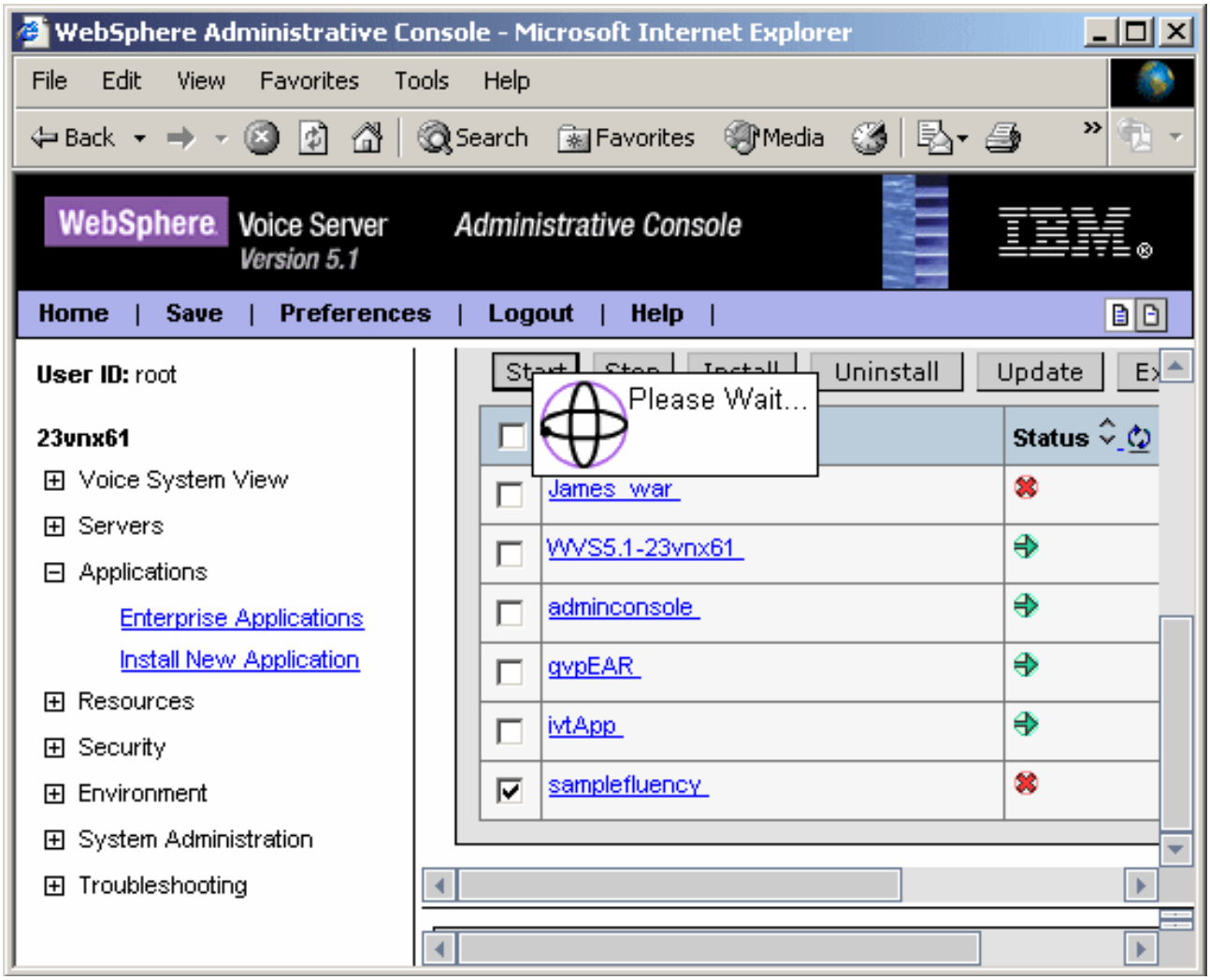

Figure 3-17 Start the Speech Application on WebSphere Application Server
Screen capture reprinted by permission from Microsoft Corporation

Step 4: Assign the channels on the IR to the URI that maps the WAR file

Once the Application has been assigned, the next step is to assign applications to the IR channels. This assignment ensures that calls presented to the IVR are routed to the correct application.

This can be accomplished via the Avaya Web Admin Tool, as shown in Figure 3-18. Detailed explanation on assigning channels is provided in 2.4.3, “Avaya IR V1.3 configuration” on page 20.

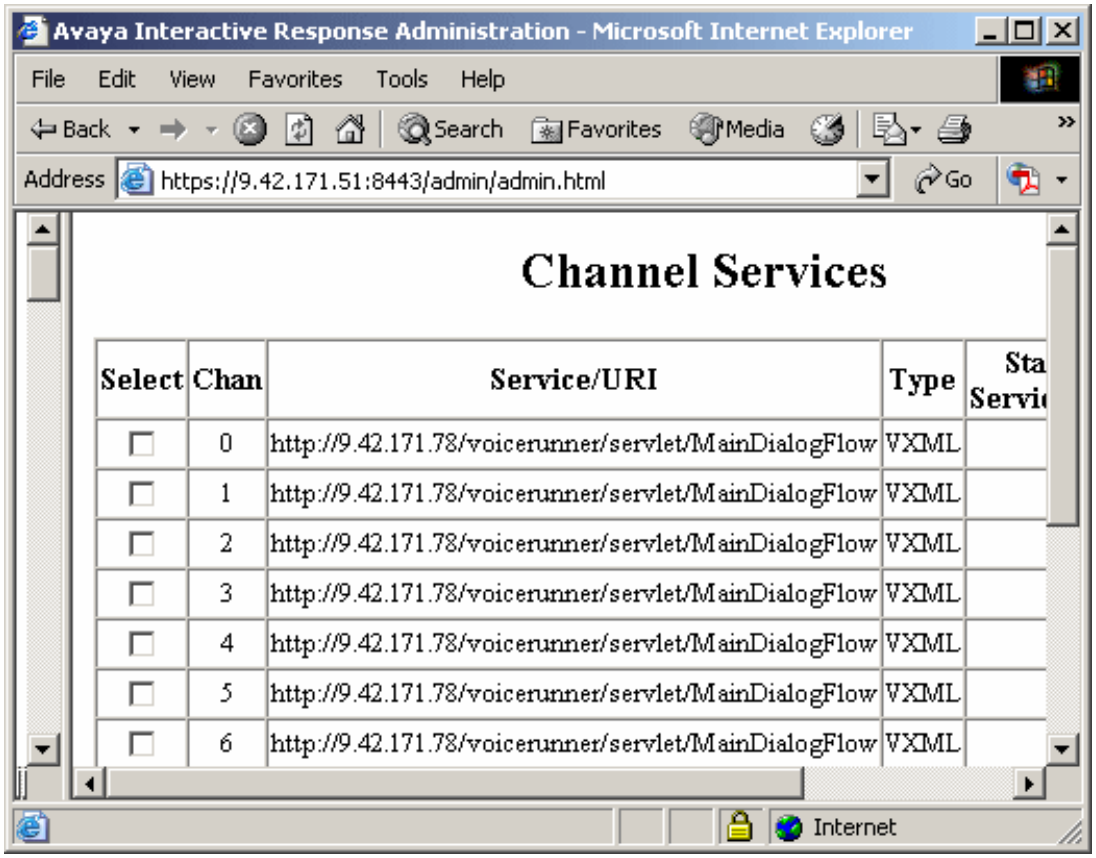

Figure 3-18 Channel Assignment on Avaya IR using Avaya Web Admin

Screen capture reprinted by permission from Microsoft Corporation

Step 5: Place a test call to validate the application

This was done using the Gordon Kapes System 930 Telephony Simulator. If the URI does not work or if the VoiceXML application is not VoiceXML complaint, Avaya IR plays a standard error message that says:

"A serious error has occurred. Exiting"

4

Testing and troubleshooting

This chapter articulates the various tests and anomalies discovered during our testing. Also, general troubleshooting tips and techniques are discussed. However, for detailed information about managing, monitoring, and debugging the WebSphere Voice Server environment, refer to the redbook *WebSphere Voice Server for Multiplatforms V5.1.1/V5.1.2 Handbook,* SG24-6447.

© Copyright IBM Corp. 2005. All rights reserved.

4.1 Testing Avaya IR

The *Avaya Interactive Response Installation, Migration, and Troubleshooting Guide* covers, in detail, how to configure and get the IR functioning. That document can be found at the following Web site:

`http://support.avaya.com`

This section complements the information available in that guide.

4.1.1 Verifying the Installation

An easy way to check the health of the IR is to go to the Admin Console and click **Report Voice System Status**, as shown in Figure 4-1 and **Display Voice Equipment Status**, as shown in Figure 4-2 on page 69.

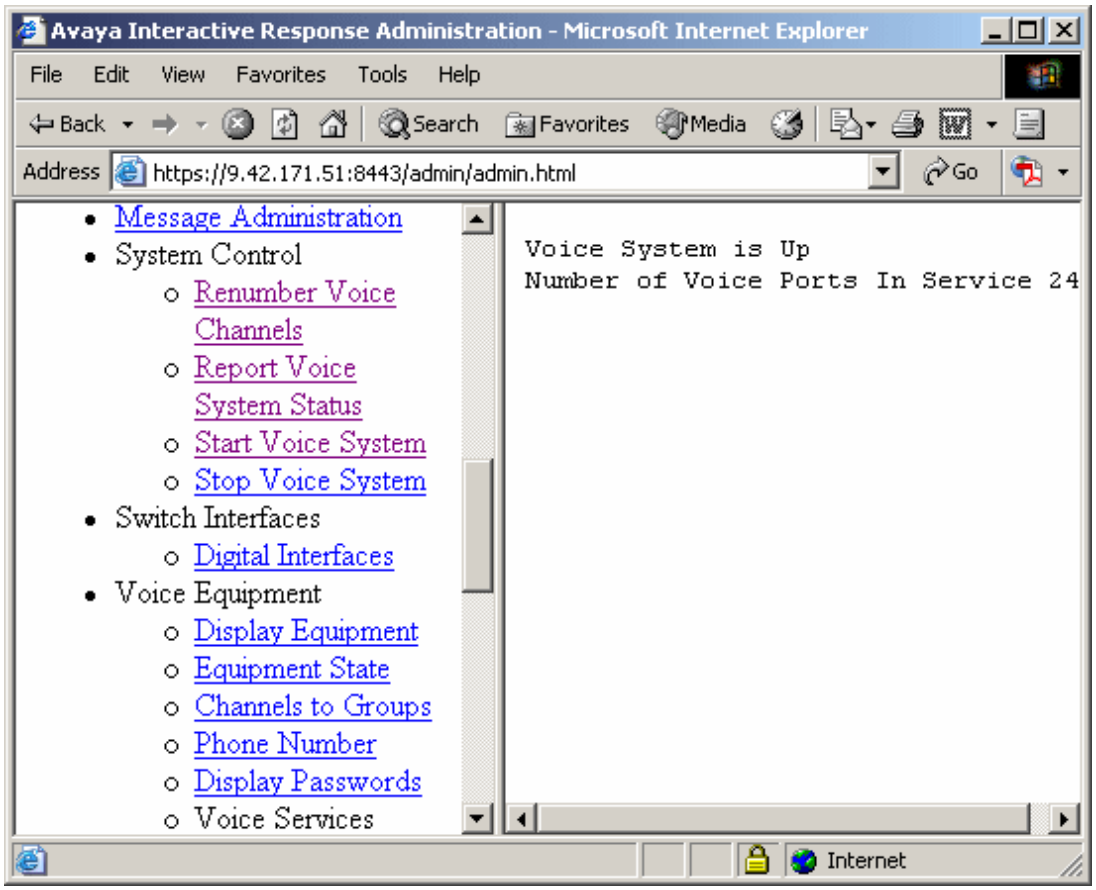

Figure 4-1 Check Voice System status on the Avaya IR
Screen capture reprinted by permission from Microsoft Corporation

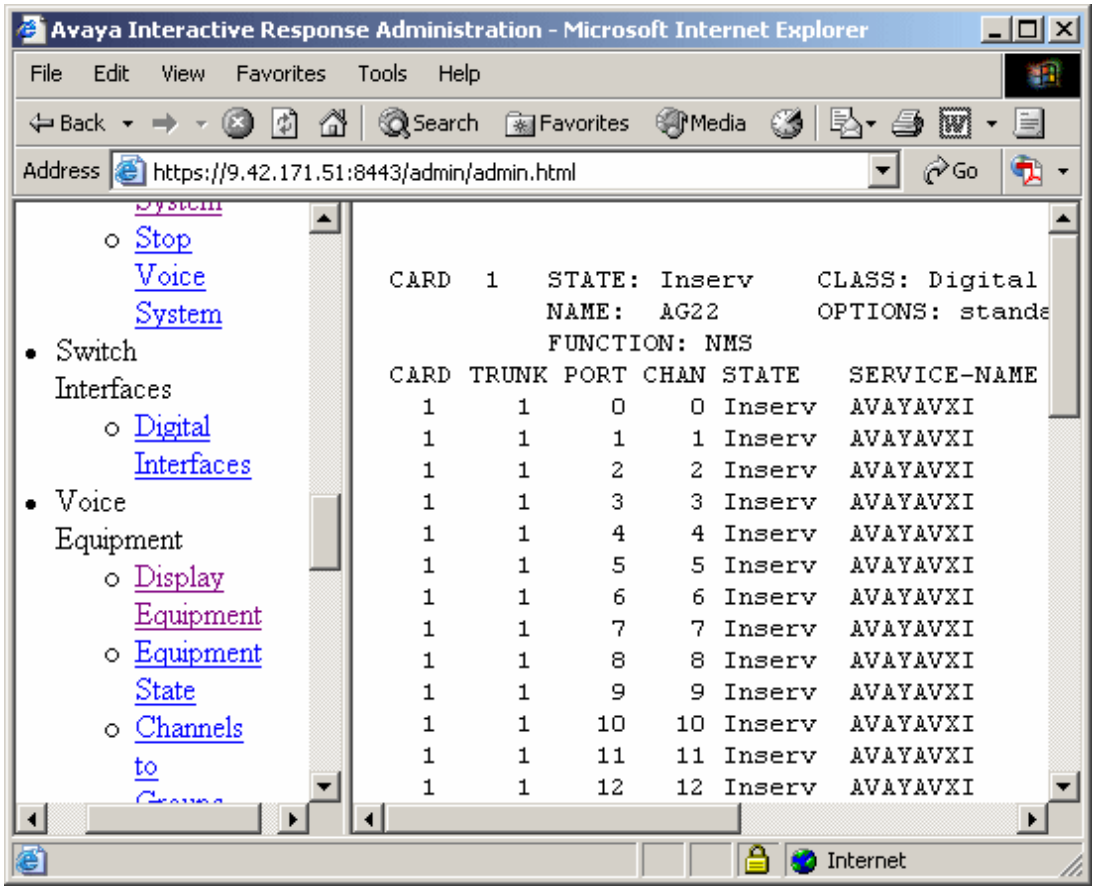
Avaya Interactive Response Administration - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address https://9.42.171.51:8443/admin/admin.html
Go
Stop Voice System
Switch Interfaces
Digital Interfaces
Voice Equipment
Display Equipment
Equipment State
Channels to
CARD 1 STATE: Inserv CLASS: Digital
NAME: AG22 OPTIONS: stande
FUNCTION: NMS
CARD TRUNK PORT CHAN STATE SERVICE-NAME
1 1 0 0 Inserv AVAYAVXI
1 1 1 1 Inserv AVAYAVXI
1 1 2 2 Inserv AVAYAVXI
1 1 3 3 Inserv AVAYAVXI
1 1 4 4 Inserv AVAYAVXI
1 1 5 5 Inserv AVAYAVXI
1 1 6 6 Inserv AVAYAVXI
1 1 7 7 Inserv AVAYAVXI
1 1 8 8 Inserv AVAYAVXI
1 1 9 9 Inserv AVAYAVXI
1 1 10 10 Inserv AVAYAVXI
1 1 11 11 Inserv AVAYAVXI
1 1 12 12 Inserv AVAYAVXI
Internet

Figure 4-2 Display Voice Equipment Status

Screen capture reprinted by permission from Microsoft Corporation

If the status of the Voice System is not Up, then click **Start Voice System**. This should start the voice system. The Avaya Admin Console provides a mechanism to start and stop the voice system, as shown in Figure 4-3.

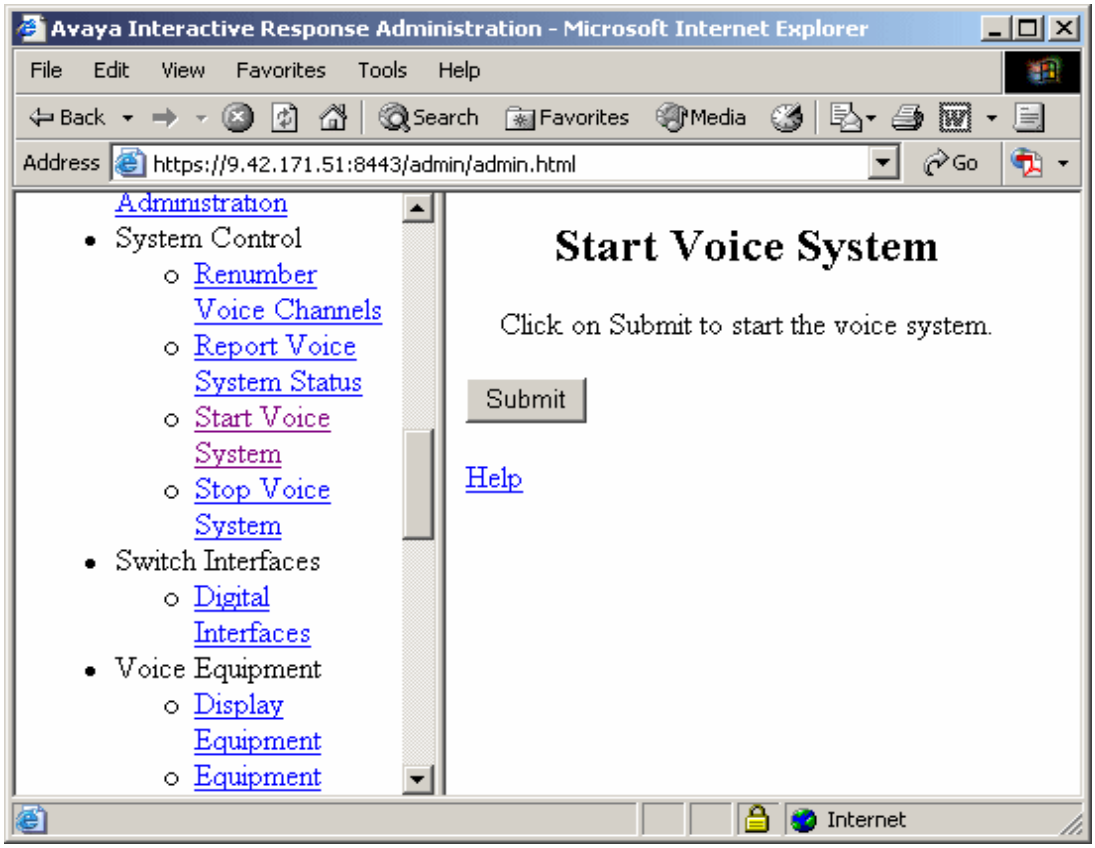

Figure 4-3 Start Avaya IR Voice System
Screen capture reprinted by permission from Microsoft Corporation

If you are still having trouble starting the voice system, check the error logs.

Note: It is important to have the valid license information configured on the IR. If the licensing is not take care, the voice system will not start.

Note: Channel State and Card State are good indicators of the equipment status. Inserv stands for inservice and is a sign of a working Avaya IR. If the card and the channel state is set to FOOS or ManualOOS state, please refer to the *Avaya Interactive Response Installation, Migration, and Troubleshooting Guide*. The guide will walk you through the steps to assign the telephony card as well as the channels into service.

4.1.2 Checking what packages are installed

You can find out what software packages have been installed on your Avaya IR system by using the **pkginfo** command. Log in as root onto the Avaya IR, open a command window, and type the following:

```
pkginfo -l > pkginfo.list
```

This will redirect the output, which can be large, to a file for easier viewing. An example of an entry for a package is shown in Example 4-1.

Example 4-1 Package entry

```
PKGINST:  AVmrcpasr
      NAME:  MRCP ASR Proxy
  CATEGORY:  IVR
      ARCH:  sparc.sun4u
   VERSION:  1.3.104
    VENDOR:  Avaya Inc.
    PSTAMP:  20050128191330
  INSTDATE:  Apr 21 2005 10:12
    STATUS:  completely installed
     FILES:       10 installed pathnames
                  10 shared pathnames
                   7 directories
                   2 executables
                33563 blocks used (approx)
```

4.1.3 Establishing telephone connectivity

Once the Avaya IR has been configured and licensed, the next step is to start the voice system. If the voice system comes up cleanly, then attempt to place a call. If the Avaya IR answers the call, you are in business. But if the Avaya IR does not answer the call, you may have a problem where the channels on the IR are not assigned to an application.

Once again, checking the card and channel state, as explained in Figure 4-2 on page 69, will help understand the current state. The way to change the card and channel state via the Avaya Web Admin tool is shown in Figure 4-4 on page 72.

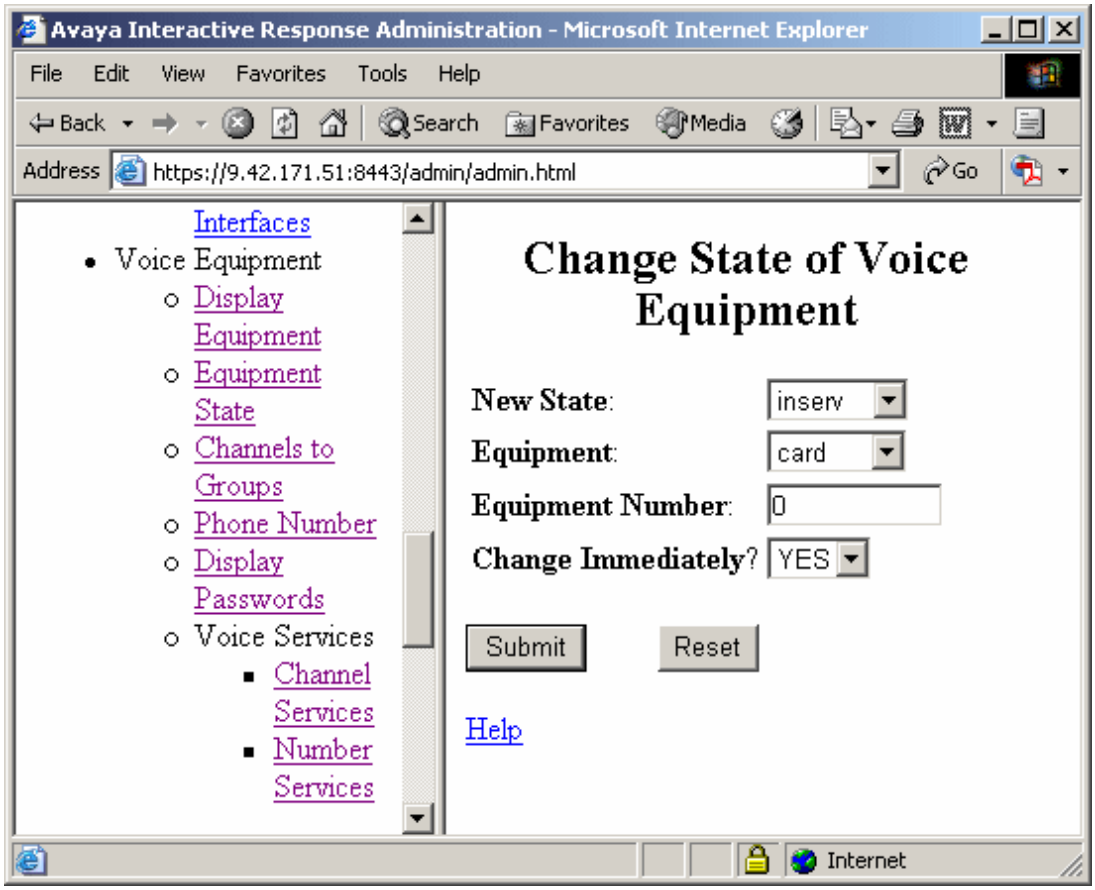

Figure 4-4 Change Card state on Avaya IR

Screen capture reprinted by permission from Microsoft Corporation

4.1.4 Managing speech resources

The Avaya IR is managed using the same Web Administration interface that we used for configuration. You can display the status of speech resources either by resource type, in our case OPSR4 and TTS0, or by the individual servers configured for each resource type. Similarly, you can manage the status of speech resources from the resource type down to specific ports on a server. Use the display status by resource type, the first one described, to get an overall view of all the speech recognition or text-to-speech resources configured.

Displaying the status of speech resources

The procedure to display the status by Resource Type is as follows:

1. Click **Display Status** from the Speech and DPR Administration group in the navigation panel of the Web Administration window.
2. The next window is titled Display Speech Proxy Status, as shown in Figure 4-5 on page 73. Click **Speech Resource Status**.

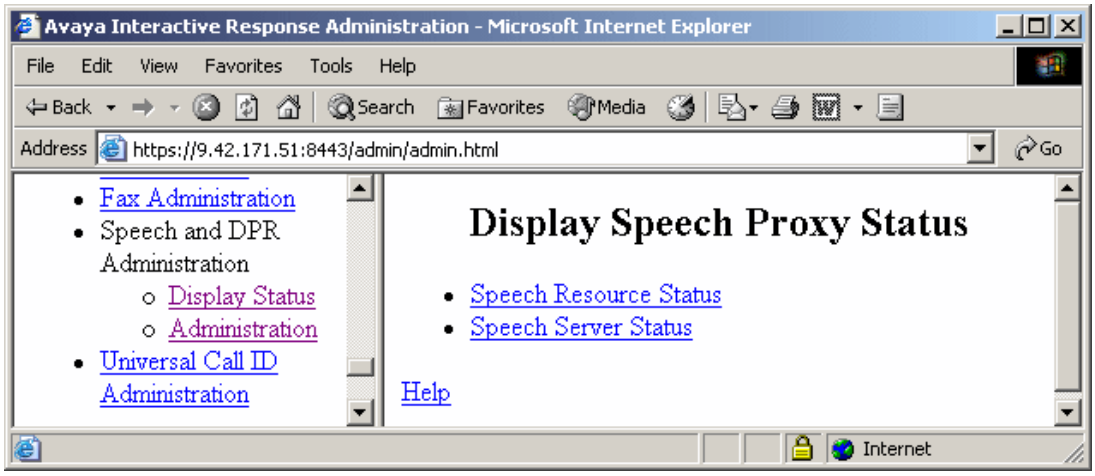

Figure 4-5 Display Speech Proxy Status window
Screen capture reprinted by permission from Microsoft Corporation

3. You will now see the Speech Resource Status window, as shown in Figure 4-6. Select **OPSR4** from the Resource Type menu list to view Speech Recognition Resources. To view Text-to-Speech Resources, select the Resource Type you configured for MRCP. In our configuration; we used **TTS0**. Now click **Submit**.

Figure 4-6 Speech Resource Status: select Resource Type
Screen capture reprinted by permission from Microsoft Corporation

4. For Speech Recognition Resources, a window similar to Figure 4-7 is displayed, and for Text-to-Speech Resources, the window will be similar to Figure 4-8 on page 75. We captured Figure 4-7 when we had the Avaya IR configured to use the Load Balancer with a cluster of three WebSphere Voice Server systems with two engines each. We captured Figure 4-8 on page 75 when we had the Avaya IR configured to use separate WebSphere Voice Server systems.

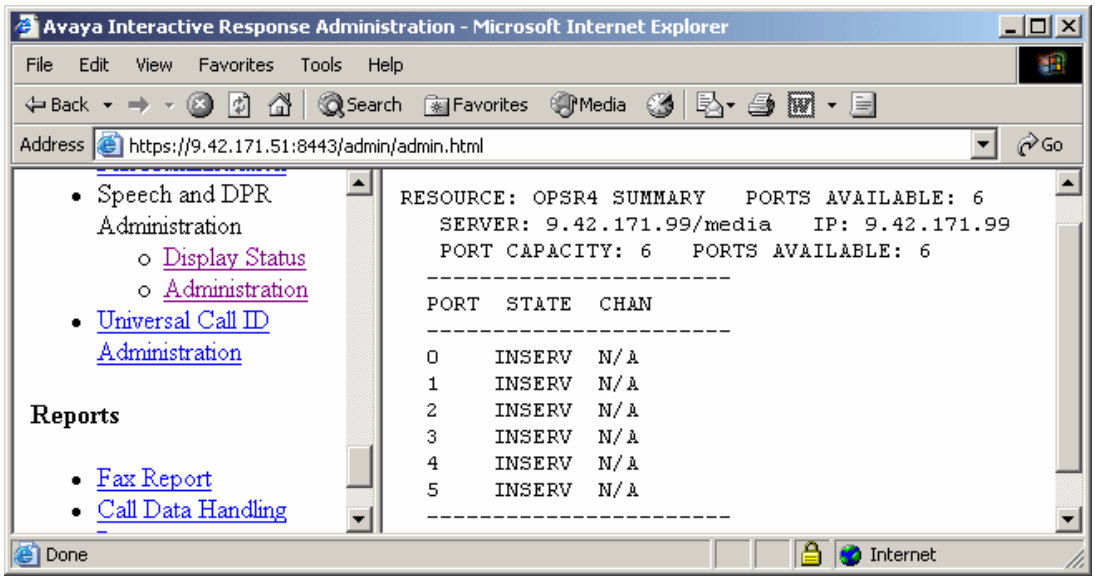

Figure 4-7 Resource Type Status Summary: Speech Recognition
Screen capture reprinted by permission from Microsoft Corporation

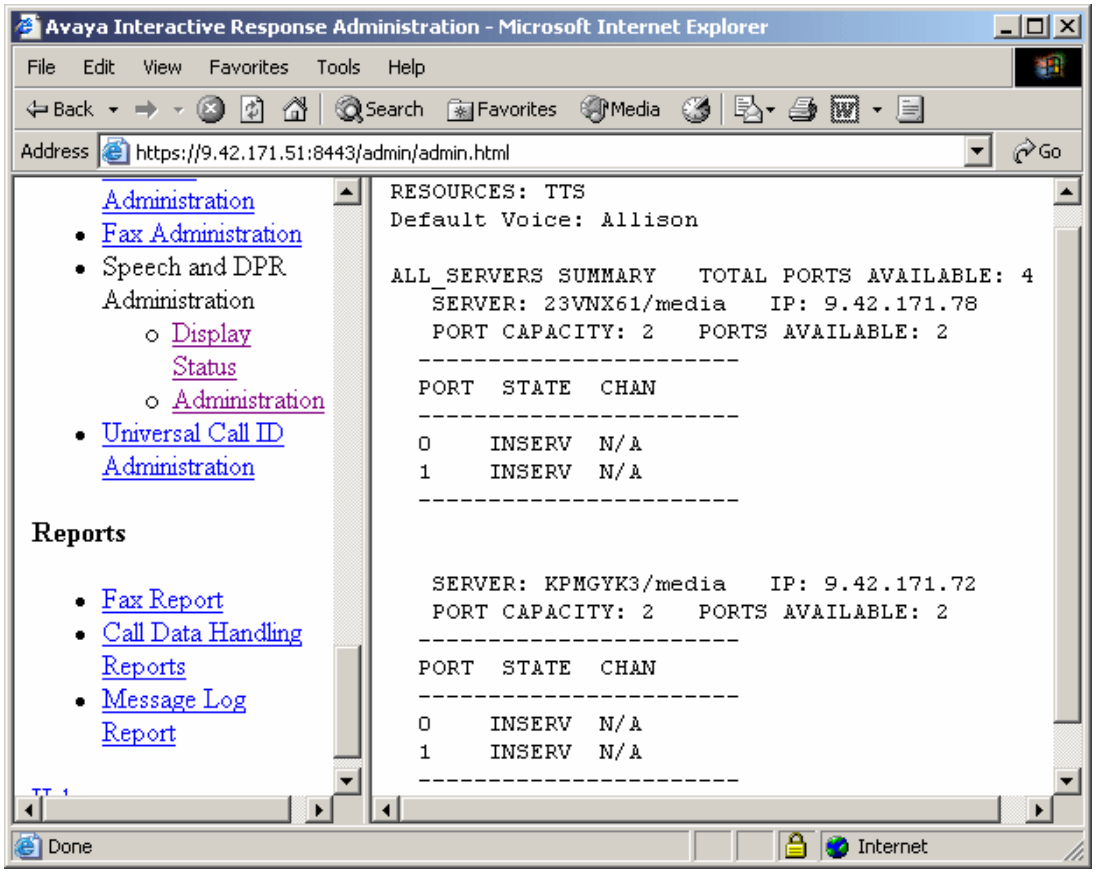

Figure 4-8 Resource Type Status Summary: Text-to-Speech
Screen capture reprinted by permission from Microsoft Corporation

Attention: We noticed that the status displayed on the Avaya IR Web Admin can sometimes be misleading. It is really an indication that the MRCP client on the Avaya IR has established a number of sessions with the MRCP server on WebSphere Voice Server. The number of available and in-service ports is *not* an indication of how many engines are running on the WebSphere Voice Server system.

5. Click **Display Status** from the Speech and DPR Administration group in the left-hand navigation of the Web Administration window to return to the Display Speech Proxy Status window.

The procedure to display the status by server is as follows:

1. From the Display Speech Proxy Status window shown in Figure 4-5 on page 73, click **Speech Server Status**.

2. You will now see the Speech Resource Status window, as shown in Figure 4-9. From the menu list, select **OPSR4** to view Speech Recognition Resources. To view Text-to-Speech Resources, select the TTS resource type you configured for MRCP. In our configuration we used **TTS0**. Now click **Submit**.

Figure 4-9 Speech Server Status: select Resource Type
Screen capture reprinted by permission from Microsoft Corporation

3. The next window will be similar to Figure 4-10. This window presents a menu list of the servers defined for the resource type you selected on the previous window. Select the server you want to display and click **Submit**.

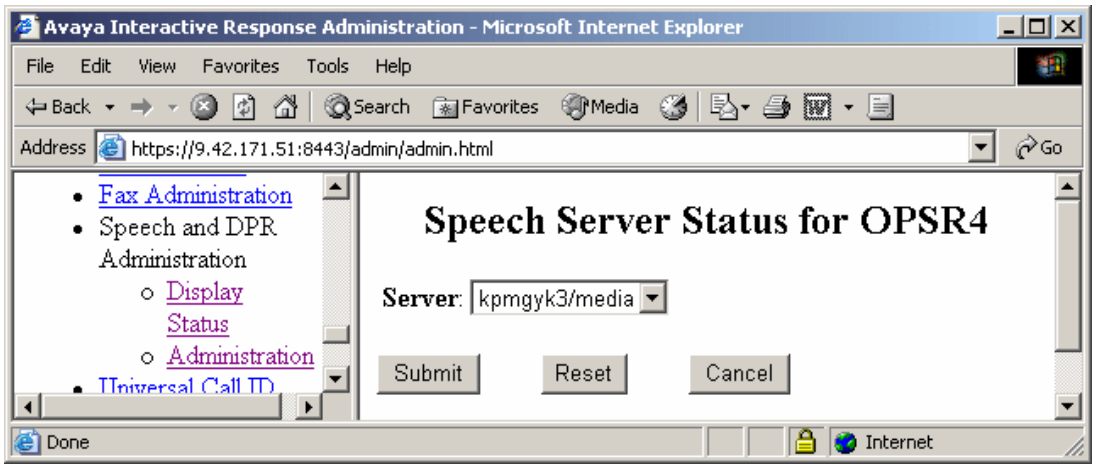

Figure 4-10 Speech Server Status: select Server
Screen capture reprinted by permission from Microsoft Corporation

4. For Speech Recognition Resources, you will see a window similar to Figure 4-11. For Text-to-Speech Resources, the window will be similar to Figure 4-12.

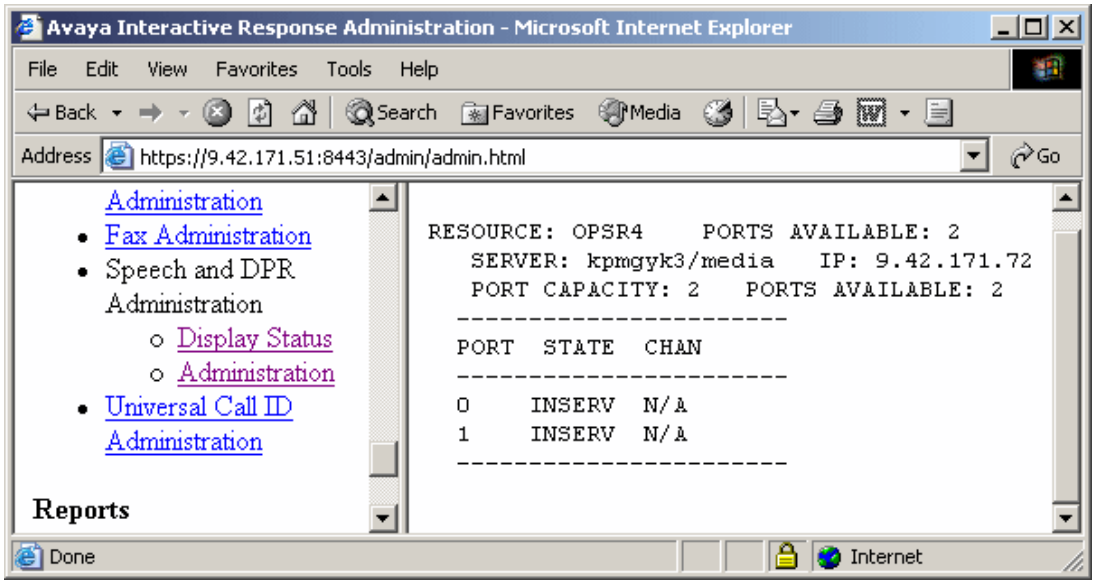

Figure 4-11 Server Resource Status Summary: Speech Recognition
Screen capture reprinted by permission from Microsoft Corporation

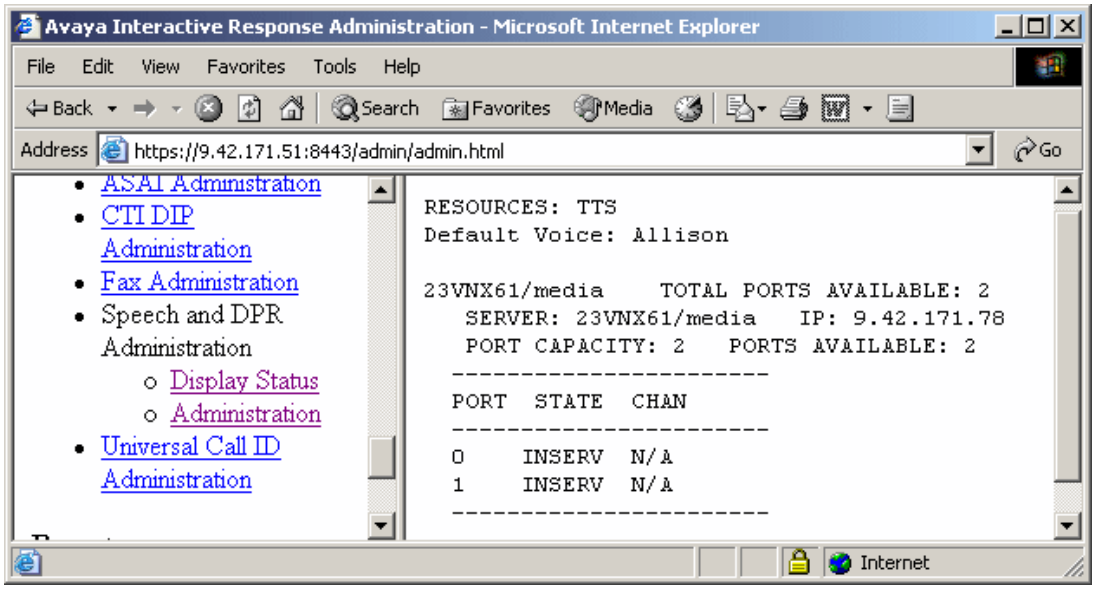

Figure 4-12 Server Resource Status Summary: Text-to-Speech
Screen capture reprinted by permission from Microsoft Corporation

5. Click **Display Status** from the Speech and DPR Administration group in the navigation panel of the Web Administration window to return to the Display Speech Proxy Status window.

Changing the status of Speech Resources

You can manage or alter the state of Speech Resources, by resource type, server, or port. This is done using the Change Speech State window shown in Figure 4-13, which is accessed by clicking **Change Speech State** on the Speech Proxy Administration window that you used when you configured the system (see Figure 2-7 on page 21).

Figure 4-13 Change Speech State window

Screen capture reprinted by permission from Microsoft Corporation

The procedure to change the state by resource is as follows:

1. Click **Change Speech Resource State** on the Change Speech State window.
2. The next window is the Change Speech Resource State window, as shown in Figure 4-14, which displays three fields.

Figure 4-14 Change Speech Resource State window

Screen capture reprinted by permission from Microsoft Corporation

The first field is the New State field, which has the options **inserv**, meaning in service, or **manoos**, meaning manually out of service. Select the state that you want the resource type to have. The second field is the Resource Type, whose state is to be changed, either to **OPSR4** or the **TTS** resource type that you configured for MRCP. The third field allows you to specify if you want the change to be made immediately or not. If you are changing the current state from in-service to out-of-service and you select **YES**, the current sessions will be terminated. Click **Submit** to make the change.

3. When the change request has been accepted, the following message is displayed:

   ```
   Change State request done
   ```

   You should check the new state by displaying the status of the resource type.

The procedure to change the state of a server is as follows:

1. Click **Change Speech Server State** on the Change Speech State window.
2. The next window is the Change Speech Resource State window, as shown in Figure 4-15. Select the resource type of the server you want to change from the menu list.

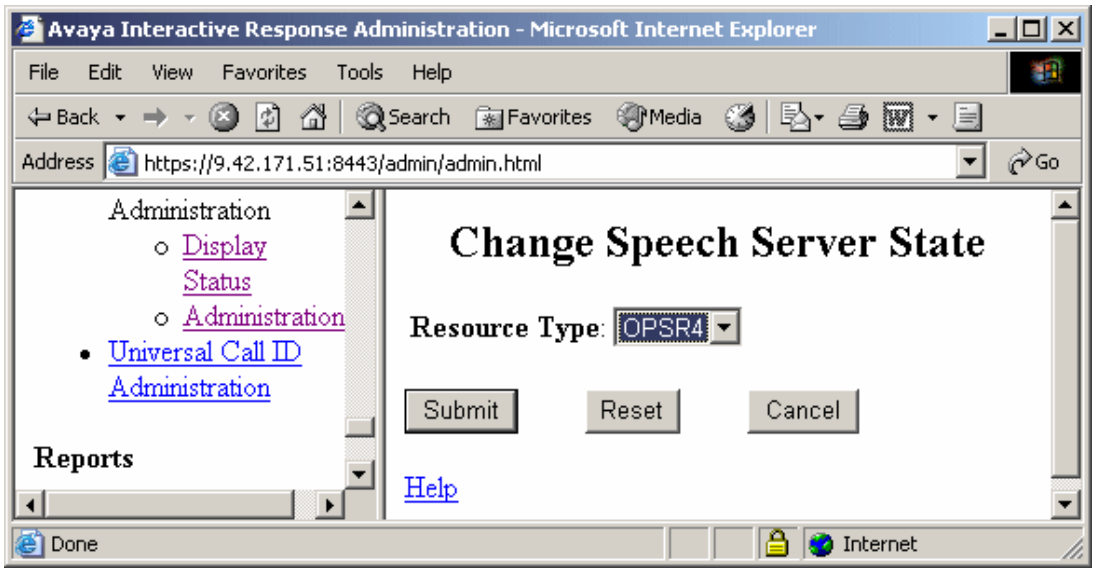

Figure 4-15 Change Speech Server State: select Resource Type
Screen capture reprinted by permission from Microsoft Corporation

3. The next window is the Change Speech Server State window, as shown in Figure 4-16. This window displays three fields.

Figure 4-16 Change Speech Server State window
Screen capture reprinted by permission from Microsoft Corporation

For the New State field, select the state that you want the server to have. For the second field, select the server whose state is to be changed. The third field allows you to specify if you want the change to be made immediately or not. If you are changing the current state from in-service to out-of-service and you select **YES**, the current sessions will be terminated. Click **Submit** to make the change.

4. When the change request has been accepted, the following message is displayed:

   Change State request done

   You should check the new state by displaying the status of the server.

The procedure to change the status of a port is as follows:

1. Click **Change Speech Resource Port State** on the Change Speech State window.
2. The next window is the Change Speech Resource Port State window, as shown in Figure 4-17 on page 81. Select the Resource Type of the port you want to change from the menu list.

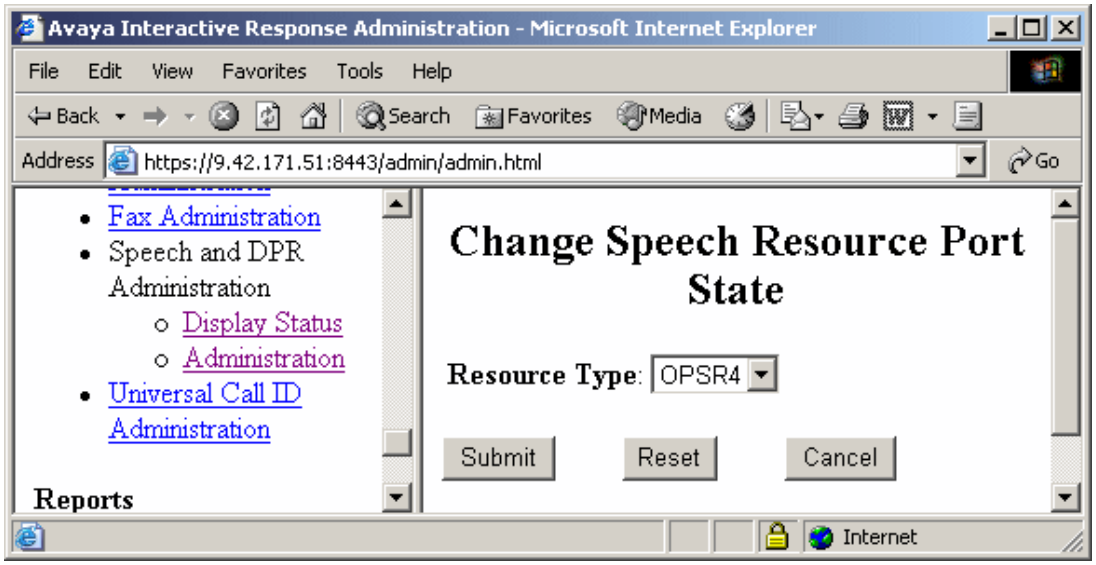

Figure 4-17 Change Speech Resource Port State: select Resource Type
Screen capture reprinted by permission from Microsoft Corporation

3. In the next window, similar to Figure 4-18, select the server name for the port you want to change from the menu list.

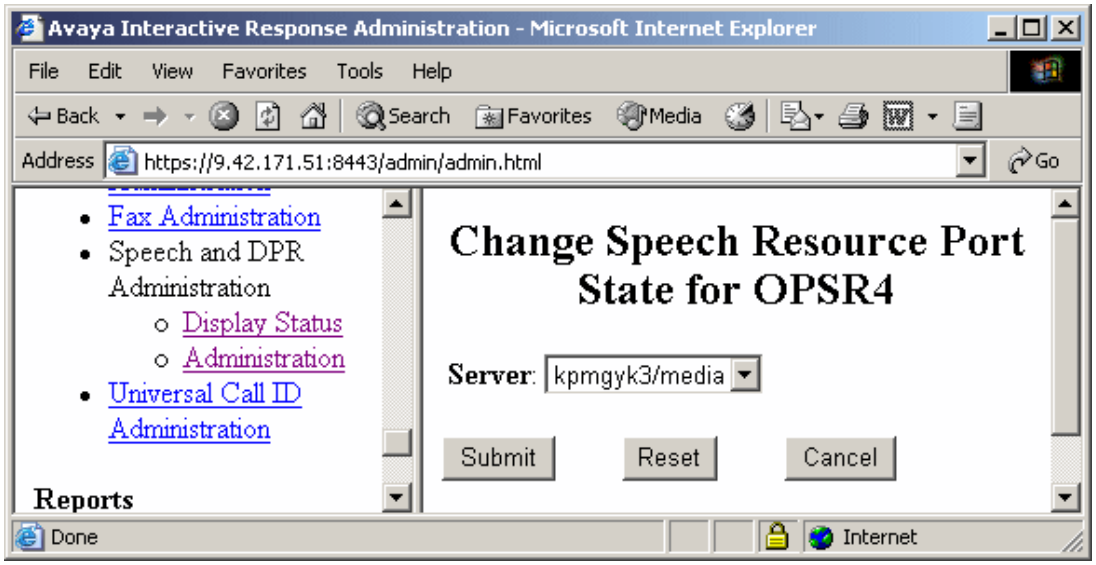

Figure 4-18 Change Speech Resource Port State: select Server
Screen capture reprinted by permission from Microsoft Corporation

4. The next window is similar to Figure 4-19. This window displays three fields.

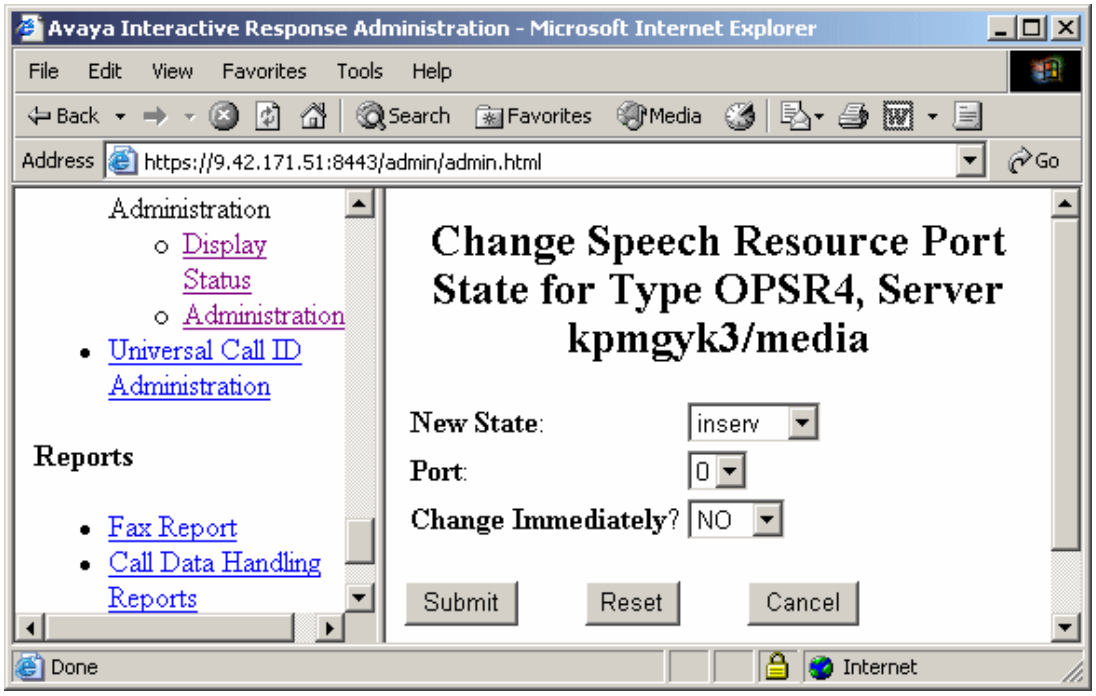

Figure 4-19 Change Speech Resource Port State window
Screen capture reprinted by permission from Microsoft Corporation

For the New State field, select the state you want the port to have. The second field is the port whose state is to be changed. The third field allows you to specify if you want the change to be made immediately or not. If you are changing the current state from in-service to out-of-service and you select **YES**, any current sessions on the port will be terminated. Click **Submit** to make the change.

5. When the change request has been accepted, the following message is displayed:

```
Change State request done
```

You should check the new state by displaying status of the port.

4.1.5 Speech Synthesis Markup Language test

Normally the voice browser and the text-to-speech engine should support the same version of the VoiceXML and Speech Synthesis Markup Language (SSML) World Wide Web Consortium (W3C) specifications. With the Avaya IR V1.3 and the IBM WebSphere Voice Server Version V5.1.2, this is *NOT* the case. Please refer to Table 4-1 for the product and its associated SSML and VoiceXML specifications.

Table 4-1 The supported version of the different systems

Product	SSML	Associated VoiceXML specification
Avaya IR V1.3	SSML V1.0 W3C Candidate Recommendation December 18, 2003	VoiceXML V2.0 W3C Candidate Recommendation February 20, 2003 (matching Avaya SSML level)
IBM WebSphere Voice Server V5.1.2	SSML V1.0 W3C Recommendation September 7, 2004	VoiceXML V2.0 W3C Recommendation March 16, 2004

Important: The Avaya AVvoicxml2-0 package enforces compliance with the VoiceXML V2.0 specification, dated March 16, 2004. The February 20, 2003 level called out above is the VoiceXML level that matches the December 18, 2003 SSML Candidate Recommendation.

Please visit the following Web site for more info:

http://www.w3.org/TR/2004/REC-speech-synthesis-20040907

A simple TTS VoiceXML application was created (see pizzassml.vxml in “Sample VoiceXML pizzassml.vxml source code” on page 89).

Not every possible VoiceXML element was tested, only a sampling. Since the results were not 100% positive, care should be taken in designing and testing VoiceXML applications that use SSML.

SSML reference information

Please refer to *WebSphere Voice Server for Multiplatforms V5.1.1/V5.1.2 Handbook,* SG24-6447 for detailed information about SSML support for WebSphere Voice Server Version 5.1.2. Please also refer to *Avaya Interactive Response Release 1.3 MRCP Feature*.

SSML testing results

This section shows a sampling of the VoiceXML elements that we tested and their results.

Emphasis element

Emphasis was tested with level=strong and level=reduced. The tester was not able to detect any difference in the audio produced, so the emphasis element appeared NOT to be effective. There was no error logged in the Avaya IR.

Voice element

Voice was tested with age=”18” and age=”6”. The tester was not able to detect any difference in the audio produced, so the Voice element appeared NOT to be effective. There was no error logged in the Avaya IR.

Prosody element

Prosody was tested with rate=”-40%”. The tester was able to detect a slowdown in the spoken audio produced, so the Prosody element appeared to function as expected.

Restriction: As stated on page 6 of the *Avaya Interactive Response Release 1.3 MRCP Feature*, the contour attribute of the prosody element is incompatible with WebSphere Voice Server Version 5.1.2. The W3C SSML specifications and WebSphere Voice Server Version 5.1.2 expects HZ. Avaya IR Version 1.3 does not use HZ.

Say-as

Say-as was tested with interpret-as=”vxml:phone”. The tester was able to detect that the spoken audio sounded like a phone number and the word extension was inserted before the 56, so the Say-as element appeared to function as expected.

4.2 Load test

We ran a load test using the Call Generator on the Gordon Kapes System 930 Telephony Simulator. The Avaya IR was connected to the Gordon Kapes System 930 Telephony Simulator with one T1 trunk running ISDN. We configured 24 speech recognition ports and 24 text-to-speech ports. We used two WebSphere Voice Server systems, one Microsoft Windows and the other Linux, each with 12 speech recognition and 12 text-to-speech engines.

Since scripts for the generated calls are very simple, we used a simple application that played two synthesized prompts, did speech recognition using a simple grammar, played another synthesized prompt, and exited. The average call duration was in the order of 12 seconds.

The call generator was configured to run 23 calls concurrently with 50 ms between calls. The test was run for approximately one hour in which time 3100 calls were handled without error on either the Avaya or WebSphere Voice Server systems.

4.3 Troubleshooting tips

In this section, we describe some of the problems and errors we encountered, and how we resolved them.

Length of document and grammar references

One of the applications we used for testing used a URI reference for a grammar that was 223 characters in length. WebSphere Voice Server was able to find and retrieve the grammar, but was unable to store the grammar in the first level cache because the fully qualified file name exceeded the 255 maximum allowed on Windows 2003.

In the SystemOut.log on the WebSphere Voice Server system, we found the message `CWVRR5004E`. This message contains the original grammar reference and the file name that the system is attempting to use to store the grammar, and the following two text messages:

```
System Message: unable to add entry <filename>
The system cannot find the path specified
```

To avoid this problem, you can do one of the following:

- Shorten the URI reference until the generated file name for the cache is less than 255 characters.
- Make the grammar non-cacheable by setting the cache-control to no-cache.
- Change the location of the caches to reduce the length of path portion of the file name. This can be done by altering the values of com.ibm.voice.server.rr.cache and com.ibm.voice.server.gm.cache parameters using the **`wsadmin`** command.

  We chose this method, and did the following:

  a. Created a C:\RRcache directory and a C:\GMcache directory.

  b. Used the following **`wsadmin`** commands:

  ```
  set wvscfg [$AdminControl queryNames *:*,type=WVSAdminConfig]
  $AdminControl invoke $wvscfg put {com.ibm.voice.server.rr.cache C\:/RRcache true}
  $AdminControl invoke $wvscfg put {com.ibm.voice.server.gm.cache C\:/GMcache true}
  ```

  c. Stopped and restarted WebSphere Voice Server to make the changes effective.

Important: If you are using **wsadmin** to make changes that only become effective after WebSphere Voice Server is re-started, you must remember to include the true parameter.

Avaya ports greater than WebSphere Voice Server engines

During a period of testing using a single WebSphere Voice Server system, we experienced an unusual problem with recognition. If we tried the application several times, sometimes it would execute correctly and other times it would not. We examined the Avaya VoiceXML log file, /voice1/vxml/log.txt to see if there were any error messages from the application of the browser. We found the message:

```
GrammarManager:: Internal Recognize - ASR resource removed
```

There were no errors on the WebSphere Voice Server. On closer investigation, we noticed that the Avaya configuration of the OPRS4 server had two ports defined and in-service, but on WebSphere Voice Server, only one speech recognition engine had been configured. We reconfigure WebSphere Voice Server to have two speech recognition engines, stopped and re-started the server, and the problem did not re-occur. We would expect a similar message would occur if the TTS ports and engines had been mismatched.

In general you need to make sure that the number of ports configured on the Avaya IR is the same or less than the number of engines configured on WebSphere Voice Server. However, if you have heavy telephony traffic, refer also to “MRCP configuration file” on page 35.

A

Sample code

This appendix provides the sample code used in testing and evaluating Avaya IR.

© Copyright IBM Corp. 2005. All rights reserved.

Sample VoiceXML pizza.vxml source code

Example 4-2 shows the VoiceXML source code generated by the Call flow manager for the Pizza Shop application.

Example 4-2 VoiceXML code generated by the Voice Toolkit for WebSphere Studio Call flow manager

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE vxml PUBLIC "-//W3C//DTD VOICEXML 2.0//EN" "vxml20-1115.dtd">
<vxml xmlns="http://www.w3.org/2001/vxml" version="2.0" xml:lang="en-US">
   <!--Call Flow Model-->
   <!--Begin Call Flow-->
   <property name="universals" value="help cancel" />
   <property name="audiomaxage" value="1" />
   <var name="mode" expr="'speech'" />
   <var name="previousResult" />
   <form id="S00060">
      <block>
         <audio src="./pizza%20audio/S00060s.wav">Welcome to the Pizza Shop.</audio>
         <goto next="#P00010" />
      </block>
   </form>
   <var name="P00010" />
   <form id="P00010">
      <field name="fP00010">
         <option value="cheese">cheese</option>
         <option value="pepperoni">pepperoni</option>
         <option value="all dressed">all dressed</option>
         <prompt cond="mode == 'speech'" bargein="true">
            <audio src="./pizza%20audio/P00010.wav">You can have a cheese pizza,
pepperoni, or all dressed?</audio>
         </prompt>
         <prompt cond="mode == 'dtmf'" bargein="true">You can have a cheese pizza,
pepperoni, or all dressed?</prompt>
         <filled>
            <assign name="P00010" expr="fP00010" />
            <assign name="previousResult" expr="fP00010" />
            <if cond="fP00010 == 'cheese'">
               <goto next="#S00020" />
               <elseif cond="fP00010 == 'pepperoni'" />
               <goto next="#S00030" />
               <elseif cond="fP00010 == 'all dressed'" />
               <goto next="#S00040" />
            </if>
         </filled>
      </field>
   </form>
   <form id="S00020">
      <block>
         cheese is nice and cheesy.
         <goto next="#P00020" />
      </block>
   </form>
   <form id="S00030">
      <block>
         the pepperoni is fresh.
         <goto next="#P00020" />
      </block>
   </form>
   <form id="S00040">
```

```
      <block>
         you must be extra hungry.
         <goto next="#P00020" />
      </block>
   </form>
   <!--End Call Flow-->
   <var name="P00020" />
   <form id="P00020">
      <field name="fP00020">
         <option value="Yes">Yes</option>
         <option value="No">No</option>
         <prompt cond="mode == 'speech'" bargein="true">Would you like to order
more?</prompt>
         <prompt cond="mode == 'dtmf'" bargein="true">Would you like to order
more?</prompt>
         <filled>
            <assign name="P00020" expr="fP00020" />
            <assign name="previousResult" expr="fP00020" />
            <if cond="fP00020 == 'Yes'">
               <goto next="#P00010" />
               <elseif cond="fP00020 == 'No'" />
               <goto next="#S00050" />
            </if>
         </filled>
      </field>
   </form>
   <form id="S00050">
      <block>
         <audio src="./pizza%20audio/S00050.wav">Please call back soon.</audio>
         <goto next="#Z00010" />
      </block>
   </form>
   <form id="Z00010">
      <block>Good Bye</block>
   </form>
</vxml>
```

Sample VoiceXML pizzassml.vxml source code

Example 4-3 shows the “pizzsml.vxml” source code used for testing.

Example 4-3 pizzsml.vxml source code listing

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE vxml PUBLIC "-//W3C//DTD VOICEXML 2.0//EN" "vxml20-1115.dtd">
<vxml xmlns="http://www.w3.org/2001/vxml" version="2.0" xml:lang="en-US">
   <meta http-equiv="Cache-Control" content="no-cache" />
   <property name="universals" value="help cancel" />
   <property name="audiomaxage" value="1" />
   <var name="mode" expr="'speech'" />
   <form id="Welcome">
      <block>
         <prompt>
            Welcome to the
            <emphasis level="strong">Pizza</emphasis>
            <break time="3s" />
            <emphasis level="reduced">shop.</emphasis>
         </prompt>
```

```
        <goto next="#Info1" />
      </block>
   </form>
   <form id="Info1">
      <block>
         <prompt>
            <voice age="18">We are open from 11am to 11pm 7 days a week.</voice>
         </prompt>
         <goto next="#Info2" />
      </block>
   </form>
   <form id="Info2">
      <block>
         <prompt>
            The specials are two pizzas for
            <prosody rate="-40%">$8.50.</prosody>
         </prompt>
         <goto next="#Info3" />
      </block>
   </form>
   <form id="Info3">
      <block>
         <prompt>
            <voice age="6">We have cheese only specials for children.</voice>
         </prompt>
         <goto next="#End" />
      </block>
   </form>
   <form id="End">
      <block>
         <prompt>
            Please call
            <say-as interpret-as="vxml:phone">8005551234x56</say-as>
         </prompt>
         <goto next="#Quit" />
      </block>
   </form>
   <form id="Quit">
      <block>Good Bye</block>
   </form>
</vxml>
```

Sample Hello World source code from Avaya SAB V1.3

Example 4-4 shows the sample Hello World VoiceXML source code used for testing.

Example 4-4 Hello World sample VoiceXML source code

```
<?xml version="1.0" encoding="UTF-8"?>
<Processes>
  <Process>
    <bo className="com.netdecisions.ccp.model.vr.bo.DialogModel" linkId="1">
      <attributes>
        <attribute value="true" name="sendEvents" />
        <attribute value="ibm.firsttest.helloworld" name="name" />
      </attributes>
      <owner>
        <Reference type="ignore" />
```

```
  </owner>
  <processVersions>
    <Reference type="link" linkId="2" />
  </processVersions>
</bo>
<bo className="com.netdecisions.ccp.model.vr.bo.DialogModelVersion" linkId="2">
  <attributes>
    <attribute name="requiredDataContainerHolder" />
    <attribute name="schemaForRequiredData" />
    <attribute value="true" name="sendEvents" />
    <attribute value="1" name="version" />
  </attributes>
  <environmentHolder>
    <Reference type="envHolder" />
  </environmentHolder>
  <mainFlow>
    <Reference type="link" linkId="3" />
  </mainFlow>
  <document>
    <Reference type="document" />
  </document>
  <process>
    <Reference type="link" linkId="1" />
  </process>
  <properties>
    <Reference type="link" linkId="15" />
  </properties>
  <allFlows>
    <Reference type="link" linkId="3" />
    <Reference type="link" linkId="16" />
  </allFlows>
  <triggers>
    <Reference type="link" linkId="26" />
  </triggers>
</bo>
<bo className="com.netdecisions.ccp.model.bp.BusinessProcessVersionFlow" linkId="3">
  <attributes>
    <attribute value="Main Flow" name="type" />
    <attribute value="ibm.firsttest.helloworld" name="name" />
  </attributes>
  <firstStep>
    <Reference type="link" linkId="4" />
  </firstStep>
  <version>
    <Reference type="link" linkId="2" />
  </version>
  <allSteps>
    <Reference type="link" linkId="4" />
    <Reference type="link" linkId="6" />
    <Reference type="link" linkId="12" />
    <Reference type="link" linkId="10" />
    <Reference type="link" linkId="8" />
  </allSteps>
</bo>
<bo className="com.netdecisions.ccp.model.bp.StepDefinition" linkId="4">
  <attributes>
    <attribute value="false" name="isMultiStep" />
    <attribute value="This is the first step in a process" name="description" />
    <attribute name="upperName" />
    <attribute value="20" name="y" />
```

```
        <attribute value="Begin Dialog" name="name" />
        <attribute value="60" name="x" />
      </attributes>
      <typeDef>
        <Reference type="componentVersion"
className="com.netdecisions.components.ccp.steps.FirstStep" version="1" />
      </typeDef>
      <repositoryComponentParameters>
        <Reference type="dbo">
          <bo className="com.netdecisions.ccp.dbo.bo.DBO" objectId="1114979949083">
            <objectId>
              <attribute value="1114979949083" name="id" className="java.lang.Long" />
            </objectId>
            <attributes>
              <attribute value="0" name="version" />
            </attributes>
            <values />
            <definition>
              <bo className="com.netdecisions.ccp.dbo.bo.DBODefinition"
objectId="1117727219690">
                <objectId>
                  <attribute value="1117727219690" name="id" className="java.lang.Long" />
                </objectId>
                <attributes>
                  <attribute value="Start Process" name="name" />
                  <attribute value="2" name="version" />
                </attributes>
                <valueDefinitions />
              </bo>
            </definition>
          </bo>
        </Reference>
      </repositoryComponentParameters>
      <processVersionFlow>
        <Reference type="link" linkId="3" />
      </processVersionFlow>
      <possibleNextSteps>
        <Reference type="link" linkId="5" />
      </possibleNextSteps>
    </bo>
    <bo className="com.netdecisions.ccp.model.bp.ConditionStepDefinition" linkId="5">
      <attributes>
        <attribute name="description" />
        <attribute value="135" name="y" />
        <attribute value="1" name="order" />
        <attribute value="156" name="x" />
      </attributes>
      <toStep>
        <Reference type="link" linkId="6" />
      </toStep>
      <fromStep>
        <Reference type="link" linkId="4" />
      </fromStep>
    </bo>
    <bo className="com.netdecisions.ccp.model.bp.StepDefinition" linkId="6">
      <attributes>
        <attribute value="false" name="isMultiStep" />
        <attribute value="Say Dialog Step " name="description" />
        <attribute name="upperName" />
        <attribute value="251" name="y" />
```

```
        <attribute value="Say Dialog Step" name="name" />
        <attribute value="253" name="x" />
      </attributes>
      <typeDef>
        <Reference type="componentVersion"
className="com.netdecisions.components.dialog.ccp.say.SayDialogStep" version="7" />
      </typeDef>
      <repositoryComponentParameters>
        <Reference type="dbo">
          <bo className="com.netdecisions.ccp.dbo.bo.DBO" objectId="1115532181520">
            <objectId>
              <attribute value="1115532181520" name="id" className="java.lang.Long" />
            </objectId>
            <attributes>
              <attribute value="0" name="version" />
            </attributes>
            <values>
              <bo
className="com.netdecisions.ccp.model.vr.dynamicsay.DynamicSayDBODataType"
objectId="1118019056624">
                <objectId>
                  <attribute value="1118019056624" name="id" className="java.lang.Long" />
                </objectId>
                <attributes>
                  <attribute value="false" name="runningUpdate" />
                  <attribute value="false" name="explicitBargeinForPrompt" />
                  <attribute value="&lt;html&gt;   &lt;head&gt;    &lt;/head&gt;
&lt;body&gt;     &lt;p&gt;       Welcome to Raleigh. This is a test. End of case study.
&lt;/p&gt;   &lt;/body&gt; &lt;/html&gt; " name="messageText" />
                  <attribute value="Message To Say" name="name" />
                </attributes>
              </bo>
              <bo className="com.netdecisions.ccp.dbo.bo.VoicePropertiesDBODataType"
objectId="1115136285488">
                <objectId>
                  <attribute value="1115136285488" name="id" className="java.lang.Long" />
                </objectId>
                <attributes>
                  <attribute name="fetchTimeout" />
                  <attribute value="dtmf voice" name="inputMode" />
                  <attribute value="none" name="universals" />
                  <attribute name="timeout" />
                  <attribute value="true" name="isFetchAudioPrompt" />
                  <attribute name="speedVsAccuracy" />
                  <attribute value="prefetch" name="audioFetchHint" />
                  <attribute name="incompleteTimeout" />
                  <attribute name="completeTimeout" />
                  <attribute value="default" name="bargeInStatusName" />
                  <attribute value="Main Voice Dialog Properties" name="name" />
                  <attribute name="fetchAudioMinimum" />
                  <attribute name="fetchAudio" />
                  <attribute name="fetchAudioDelay" />
                  <attribute name="sensitivity" />
                  <attribute name="confidenceLevel" />
                </attributes>
              </bo>
            </values>
            <definition>
              <bo className="com.netdecisions.ccp.dbo.bo.DBODefinition"
objectId="1117065745296">
```

```
                <objectId>
                  <attribute value="1117065745296" name="id" className="java.lang.Long" />
                </objectId>
                <attributes>
                  <attribute value="Say Dialog Step" name="name" />
                  <attribute value="2" name="version" />
                </attributes>
                <valueDefinitions>
                  <bo className="com.netdecisions.ccp.dbo.bo.DBOValueDefinition"
objectId="1115900743408">
                    <objectId>
                     <attribute value="1115900743408" name="id" className="java.lang.Long"
/>
                    </objectId>
                    <attributes>
                      <attribute name="securityPermissionRequired" />
                      <attribute value="true" name="required" />
                      <attribute value="Prompts" name="page" />
                      <attribute value="The message to be said" name="displayText" />
                      <attribute value="false" name="arrayType" />
                      <attribute
value="com.netdecisions.ccp.model.vr.dynamicsay.DynamicSayDBODataType"
name="dataTypeClassName" />
                      <attribute value="Message To Say" name="name" />
                    </attributes>
                    <defaultValue />
                    <validValues />
                  </bo>
                  <bo className="com.netdecisions.ccp.dbo.bo.DBOValueDefinition"
objectId="1117960897904">
                    <objectId>
                     <attribute value="1117960897904" name="id" className="java.lang.Long"
/>
                    </objectId>
                    <attributes>
                      <attribute name="securityPermissionRequired" />
                      <attribute value="false" name="required" />
                      <attribute value="Prompts" name="page" />
                      <attribute value="Main Voice Dialog Properties" name="displayText" />
                      <attribute value="false" name="arrayType" />
                      <attribute
value="com.netdecisions.ccp.dbo.bo.VoicePropertiesDBODataType" name="dataTypeClassName" />
                      <attribute value="Main Voice Dialog Properties" name="name" />
                    </attributes>
                    <defaultValue />
                    <validValues />
                  </bo>
                </valueDefinitions>
              </bo>
            </definition>
          </bo>
        </Reference>
      </repositoryComponentParameters>
      <processVersionFlow>
        <Reference type="link" linkId="3" />
      </processVersionFlow>
      <possibleNextSteps>
        <Reference type="link" linkId="7" />
        <Reference type="link" linkId="13" />
      </possibleNextSteps>
```

```
      <leadingSteps>
        <Reference type="link" linkId="5" />
      </leadingSteps>
    </bo>
    <bo className="com.netdecisions.ccp.model.bp.ConditionStepDefinition" linkId="7">
      <attributes>
        <attribute name="description" />
        <attribute value="296" name="y" />
        <attribute value="2" name="order" />
        <attribute value="323" name="x" />
      </attributes>
      <toStep>
        <Reference type="link" linkId="8" />
      </toStep>
      <fromStep>
        <Reference type="link" linkId="6" />
      </fromStep>
    </bo>
    <bo className="com.netdecisions.ccp.model.bp.StepDefinition" linkId="8">
      <attributes>
        <attribute value="false" name="isMultiStep" />
        <attribute value="Path out of a dialog component.  This component must follow a
dialog component" name="description" />
        <attribute name="upperName" />
        <attribute value="356" name="y" />
        <attribute value="Call Ended Response" name="name" />
        <attribute value="187" name="x" />
      </attributes>
      <typeDef>
        <Reference type="componentVersion"
className="com.netdecisions.components.dialog.ccp_components.DialogResultPath" version="2"
/>
      </typeDef>
      <repositoryComponentParameters>
        <Reference type="dbo">
          <bo className="com.netdecisions.ccp.dbo.bo.DBO" objectId="1115893111327">
            <objectId>
              <attribute value="1115893111327" name="id" className="java.lang.Long" />
            </objectId>
            <attributes>
              <attribute value="0" name="version" />
            </attributes>
            <values>
              <bo className="com.netdecisions.ccp.dbo.bo.StringDBODataType"
objectId="1116931167359">
                <objectId>
                  <attribute value="1116931167359" name="id" className="java.lang.Long" />
                </objectId>
                <attributes>
                  <attribute name="value" />
                  <attribute value="CONTROL_CLASS" name="name" />
                  <attribute value="com.fluencyvoice.runner.core.Dialog$CallEndedResponse"
name="stringValue" />
                </attributes>
              </bo>
            </values>
            <definition>
              <bo className="com.netdecisions.ccp.dbo.bo.DBODefinition"
objectId="1117231895679">
                <objectId>
```

```
                  <attribute value="1117231895679" name="id" className="java.lang.Long" />
                </objectId>
                <attributes>
                  <attribute value="Dialog Result Path" name="name" />
                  <attribute value="2" name="version" />
                </attributes>
                <valueDefinitions>
                  <bo className="com.netdecisions.ccp.dbo.bo.DBOValueDefinition"
objectId="1117625377823">
                    <objectId>
                     <attribute value="1117625377823" name="id" className="java.lang.Long"
/>
                    </objectId>
                    <attributes>
                      <attribute name="securityPermissionRequired" />
                      <attribute value="true" name="required" />
                      <attribute name="page" />
                      <attribute value="CONTROL_CLASS" name="displayText" />
                      <attribute value="false" name="arrayType" />
                      <attribute value="com.netdecisions.ccp.dbo.bo.StringDBODataType"
name="dataTypeClassName" />
                      <attribute value="CONTROL_CLASS" name="name" />
                    </attributes>
                    <defaultValue />
                    <validValues />
                  </bo>
                </valueDefinitions>
              </bo>
            </definition>
          </bo>
        </Reference>
      </repositoryComponentParameters>
      <processVersionFlow>
        <Reference type="link" linkId="3" />
      </processVersionFlow>
      <possibleNextSteps>
        <Reference type="link" linkId="9" />
      </possibleNextSteps>
      <leadingSteps>
        <Reference type="link" linkId="7" />
      </leadingSteps>
    </bo>
    <bo className="com.netdecisions.ccp.model.bp.ConditionStepDefinition" linkId="9">
      <attributes>
        <attribute name="description" />
        <attribute value="428" name="y" />
        <attribute value="1" name="order" />
        <attribute value="343" name="x" />
      </attributes>
      <toStep>
        <Reference type="link" linkId="10" />
      </toStep>
      <fromStep>
        <Reference type="link" linkId="8" />
      </fromStep>
    </bo>
    <bo className="com.netdecisions.ccp.model.bp.StepDefinition" linkId="10">
      <attributes>
        <attribute value="false" name="isMultiStep" />
```

```
        <attribute value="This step is used as a marker to represent the last step in a
process flow." name="description" />
        <attribute name="upperName" />
        <attribute value="500" name="y" />
        <attribute value="End Dialog" name="name" />
        <attribute value="500" name="x" />
      </attributes>
      <typeDef>
        <Reference type="componentVersion"
className="com.netdecisions.components.ccp.steps.DefaultTerminalStep" version="1" />
      </typeDef>
      <repositoryComponentParameters>
        <Reference type="dbo">
          <bo className="com.netdecisions.ccp.dbo.bo.DBO" objectId="1118190407018">
            <objectId>
              <attribute value="1118190407018" name="id" className="java.lang.Long" />
            </objectId>
            <attributes>
              <attribute value="0" name="version" />
            </attributes>
            <values />
            <definition>
              <bo className="com.netdecisions.ccp.dbo.bo.DBODefinition"
objectId="1117403399658">
                <objectId>
                  <attribute value="1117403399658" name="id" className="java.lang.Long" />
                </objectId>
                <attributes>
                  <attribute value="End Process" name="name" />
                  <attribute value="2" name="version" />
                </attributes>
                <valueDefinitions />
              </bo>
            </definition>
          </bo>
        </Reference>
      </repositoryComponentParameters>
      <processVersionFlow>
        <Reference type="link" linkId="3" />
      </processVersionFlow>
      <leadingSteps>
        <Reference type="link" linkId="11" />
        <Reference type="link" linkId="9" />
      </leadingSteps>
    </bo>
    <bo className="com.netdecisions.ccp.model.bp.ConditionStepDefinition" linkId="11">
      <attributes>
        <attribute name="description" />
        <attribute value="414" name="y" />
        <attribute value="1" name="order" />
        <attribute value="343" name="x" />
      </attributes>
      <toStep>
        <Reference type="link" linkId="10" />
      </toStep>
      <fromStep>
        <Reference type="link" linkId="12" />
      </fromStep>
    </bo>
    <bo className="com.netdecisions.ccp.model.bp.StepDefinition" linkId="12">
```

```
      <attributes>
        <attribute value="false" name="isMultiStep" />
        <attribute value="Path out of a dialog component.  This component must follow a
dialog component" name="description" />
        <attribute name="upperName" />
        <attribute value="329" name="y" />
        <attribute value="Completed Response" name="name" />
        <attribute value="187" name="x" />
      </attributes>
      <typeDef>
        <Reference type="componentVersion"
className="com.netdecisions.components.dialog.ccp_components.DialogResultPath" version="2"
/>
      </typeDef>
      <repositoryComponentParameters>
        <Reference type="dbo">
          <bo className="com.netdecisions.ccp.dbo.bo.DBO" objectId="1117233233919">
            <objectId>
              <attribute value="1117233233919" name="id" className="java.lang.Long" />
            </objectId>
            <attributes>
              <attribute value="0" name="version" />
            </attributes>
            <values>
              <bo className="com.netdecisions.ccp.dbo.bo.StringDBODataType"
objectId="1118617378015">
                <objectId>
                  <attribute value="1118617378015" name="id" className="java.lang.Long" />
                </objectId>
                <attributes>
                  <attribute name="value" />
                  <attribute value="CONTROL_CLASS" name="name" />
                  <attribute
value="com.fluencyvoice.runner.dialog.voice.basic.MessageDialog$CompletedResponse"
name="stringValue" />
                </attributes>
              </bo>
            </values>
            <definition>
              <bo className="com.netdecisions.ccp.dbo.bo.DBODefinition"
objectId="1117657799199">
                <objectId>
                  <attribute value="1117657799199" name="id" className="java.lang.Long" />
                </objectId>
                <attributes>
                  <attribute value="Dialog Result Path" name="name" />
                  <attribute value="2" name="version" />
                </attributes>
                <valueDefinitions>
                  <bo className="com.netdecisions.ccp.dbo.bo.DBOValueDefinition"
objectId="1117041956031">
                    <objectId>
                      <attribute value="1117041956031" name="id" className="java.lang.Long"
/>
                    </objectId>
                    <attributes>
                      <attribute name="securityPermissionRequired" />
                      <attribute value="true" name="required" />
                      <attribute name="page" />
                      <attribute value="CONTROL_CLASS" name="displayText" />
```

```
                    <attribute value="false" name="arrayType" />
                    <attribute value="com.netdecisions.ccp.dbo.bo.StringDBODataType"
name="dataTypeClassName" />
                    <attribute value="CONTROL_CLASS" name="name" />
                  </attributes>
                  <defaultValue />
                  <validValues />
                </bo>
              </valueDefinitions>
            </bo>
          </definition>
        </bo>
      </Reference>
    </repositoryComponentParameters>
    <processVersionFlow>
      <Reference type="link" linkId="3" />
    </processVersionFlow>
    <possibleNextSteps>
      <Reference type="link" linkId="11" />
    </possibleNextSteps>
    <leadingSteps>
      <Reference type="link" linkId="13" />
    </leadingSteps>
  </bo>
  <bo className="com.netdecisions.ccp.model.bp.ConditionStepDefinition" linkId="13">
    <attributes>
      <attribute name="description" />
      <attribute value="251" name="y" />
      <attribute value="1" name="order" />
      <attribute value="323" name="x" />
    </attributes>
    <toStep>
      <Reference type="link" linkId="12" />
    </toStep>
    <ruleDefinition>
      <Reference type="link" linkId="14" />
    </ruleDefinition>
    <fromStep>
      <Reference type="link" linkId="6" />
    </fromStep>
  </bo>
  <bo className="com.netdecisions.ccp.model.rule.RuleDefinition" linkId="14">
    <attributes>
      <attribute value="false" name="isTrueBranch" />
      <attribute value="Evaluates the last control type" name="description" />
      <attribute name="upperName" />
      <attribute value="Control Type Rule" name="name" />
    </attributes>
    <typeDef>
      <Reference type="componentVersion"
className="com.netdecisions.components.dialog.ccp_components.ControlTypeRule" version="3"
/>
    </typeDef>
    <repositoryComponentParameters>
      <Reference type="dbo">
        <bo className="com.netdecisions.ccp.dbo.bo.DBO" objectId="1115725907375">
          <objectId>
            <attribute value="1115725907375" name="id" className="java.lang.Long" />
          </objectId>
          <attributes>
```

```
              <attribute value="0" name="version" />
            </attributes>
            <values>
              <bo className="com.netdecisions.ccp.dbo.bo.StringDBODataType"
objectId="1117714903727">
                <objectId>
                  <attribute value="1117714903727" name="id" className="java.lang.Long" />
                </objectId>
                <attributes>
                  <attribute name="value" />
                  <attribute value="CONTROL_CLASS" name="name" />
                  <attribute
value="com.fluencyvoice.runner.dialog.voice.basic.MessageDialog$CompletedResponse"
name="stringValue" />
                </attributes>
              </bo>
              <bo className="com.netdecisions.ccp.dbo.bo.IntegerDBODataType"
objectId="1117881382223">
                <objectId>
                  <attribute value="1117881382223" name="id" className="java.lang.Long" />
                </objectId>
                <attributes>
                  <attribute value="100" name="value" />
                  <attribute value="CONTROL_CLASS_DOCUMENT_VERSION" name="name" />
                </attributes>
              </bo>
            </values>
            <definition>
              <bo className="com.netdecisions.ccp.dbo.bo.DBODefinition"
objectId="1115384875279">
                <objectId>
                  <attribute value="1115384875279" name="id" className="java.lang.Long" />
                </objectId>
                <attributes>
                  <attribute value="Control Type Rule" name="name" />
                  <attribute value="2" name="version" />
                </attributes>
                <valueDefinitions>
                  <bo className="com.netdecisions.ccp.dbo.bo.DBOValueDefinition"
objectId="1116496077967">
                    <objectId>
                     <attribute value="1116496077967" name="id" className="java.lang.Long"
/>
                    </objectId>
                    <attributes>
                      <attribute name="securityPermissionRequired" />
                      <attribute value="true" name="required" />
                      <attribute name="page" />
                      <attribute value="CONTROL_CLASS" name="displayText" />
                      <attribute value="false" name="arrayType" />
                      <attribute value="com.netdecisions.ccp.dbo.bo.StringDBODataType"
name="dataTypeClassName" />
                      <attribute value="CONTROL_CLASS" name="name" />
                    </attributes>
                    <defaultValue />
                    <validValues />
                  </bo>
                  <bo className="com.netdecisions.ccp.dbo.bo.DBOValueDefinition"
objectId="1116115068527">
                    <objectId>
```

```
                      <attribute value="1116115068527" name="id" className="java.lang.Long"
/>
                    </objectId>
                    <attributes>
                      <attribute name="securityPermissionRequired" />
                      <attribute value="false" name="required" />
                      <attribute name="page" />
                      <attribute value="Control class document version" name="displayText"
/>
                      <attribute value="false" name="arrayType" />
                      <attribute value="com.netdecisions.ccp.dbo.bo.IntegerDBODataType"
name="dataTypeClassName" />
                      <attribute value="CONTROL_CLASS_DOCUMENT_VERSION" name="name" />
                    </attributes>
                    <defaultValue />
                    <validValues />
                  </bo>
                </valueDefinitions>
              </bo>
            </definition>
          </bo>
        </Reference>
      </repositoryComponentParameters>
    </bo>
    <bo className="com.netdecisions.ccp.model.vr.bo.DialogModelVersionProperties"
linkId="15">
      <attributes>
        <attribute value="3" name="maxErrorsPermitted" />
        <attribute value="50" name="defaultConfidenceLevel" />
        <attribute value="true" name="bargeInAllowed" />
        <attribute value="3000" name="noInputTimeout" />
      </attributes>
    </bo>
    <bo className="com.netdecisions.ccp.model.bp.BusinessProcessVersionFlow" linkId="16">
      <attributes>
        <attribute value="Unhandled Control" name="type" />
        <attribute value="Default Unhandled Control Trigger" name="name" />
      </attributes>
      <firstStep>
        <Reference type="link" linkId="17" />
      </firstStep>
      <version>
        <Reference type="link" linkId="2" />
      </version>
      <allSteps>
        <Reference type="link" linkId="17" />
        <Reference type="link" linkId="23" />
        <Reference type="link" linkId="21" />
        <Reference type="link" linkId="19" />
      </allSteps>
    </bo>
    <bo className="com.netdecisions.ccp.model.bp.StepDefinition" linkId="17">
      <attributes>
        <attribute value="false" name="isMultiStep" />
        <attribute value="This is the first step in a process" name="description" />
        <attribute name="upperName" />
        <attribute value="20" name="y" />
        <attribute value="Start Process" name="name" />
        <attribute value="60" name="x" />
      </attributes>
```

```
      <typeDef>
        <Reference type="componentVersion"
className="com.netdecisions.components.ccp.steps.FirstStep" version="1" />
      </typeDef>
      <repositoryComponentParameters>
        <Reference type="dbo">
          <bo className="com.netdecisions.ccp.dbo.bo.DBO" objectId="1118056733578">
            <objectId>
              <attribute value="1118056733578" name="id" className="java.lang.Long" />
            </objectId>
            <attributes>
              <attribute value="0" name="version" />
            </attributes>
            <values />
            <definition>
              <bo className="com.netdecisions.ccp.dbo.bo.DBODefinition"
objectId="1115676205738">
                <objectId>
                  <attribute value="1115676205738" name="id" className="java.lang.Long" />
                </objectId>
                <attributes>
                  <attribute value="Start Process" name="name" />
                  <attribute value="2" name="version" />
                </attributes>
                <valueDefinitions />
              </bo>
            </definition>
          </bo>
        </Reference>
      </repositoryComponentParameters>
      <processVersionFlow>
        <Reference type="link" linkId="16" />
      </processVersionFlow>
      <possibleNextSteps>
        <Reference type="link" linkId="18" />
        <Reference type="link" linkId="24" />
      </possibleNextSteps>
    </bo>
    <bo className="com.netdecisions.ccp.model.bp.ConditionStepDefinition" linkId="18">
      <attributes>
        <attribute name="description" />
        <attribute value="282" name="y" />
        <attribute value="2" name="order" />
        <attribute value="52" name="x" />
      </attributes>
      <toStep>
        <Reference type="link" linkId="19" />
      </toStep>
      <fromStep>
        <Reference type="link" linkId="17" />
      </fromStep>
    </bo>
    <bo className="com.netdecisions.ccp.model.bp.StepDefinition" linkId="19">
      <attributes>
        <attribute value="false" name="isMultiStep" />
        <attribute value="This step will search through the parent steps and re-run the
first dialog step that it finds." name="description" />
        <attribute name="upperName" />
        <attribute value="252" name="y" />
        <attribute value="Re-run Last Dialog Step" name="name" />
```

```
        <attribute value="220" name="x" />
      </attributes>
      <typeDef>
        <Reference type="componentVersion"
className="com.netdecisions.components.dialog.ccp_components.ReRunLastDialogStep"
version="1" />
      </typeDef>
      <repositoryComponentParameters>
        <Reference type="dbo">
          <bo className="com.netdecisions.ccp.dbo.bo.DBO" objectId="1114751904618">
            <objectId>
              <attribute value="1114751904618" name="id" className="java.lang.Long" />
            </objectId>
            <attributes>
              <attribute value="0" name="version" />
            </attributes>
            <values />
            <definition>
              <bo className="com.netdecisions.ccp.dbo.bo.DBODefinition"
objectId="1114824492746">
                <objectId>
                  <attribute value="1114824492746" name="id" className="java.lang.Long" />
                </objectId>
                <attributes>
                  <attribute value="Re-run Last Dialog Step" name="name" />
                  <attribute value="2" name="version" />
                </attributes>
                <valueDefinitions />
              </bo>
            </definition>
          </bo>
        </Reference>
      </repositoryComponentParameters>
      <processVersionFlow>
        <Reference type="link" linkId="16" />
      </processVersionFlow>
      <possibleNextSteps>
        <Reference type="link" linkId="20" />
      </possibleNextSteps>
      <leadingSteps>
        <Reference type="link" linkId="18" />
      </leadingSteps>
    </bo>
    <bo className="com.netdecisions.ccp.model.bp.ConditionStepDefinition" linkId="20">
      <attributes>
        <attribute name="description" />
        <attribute value="179" name="y" />
        <attribute value="1" name="order" />
        <attribute value="367" name="x" />
      </attributes>
      <toStep>
        <Reference type="link" linkId="21" />
      </toStep>
      <fromStep>
        <Reference type="link" linkId="19" />
      </fromStep>
    </bo>
    <bo className="com.netdecisions.ccp.model.bp.StepDefinition" linkId="21">
      <attributes>
        <attribute value="false" name="isMultiStep" />
```

```
        <attribute value="This step is used as a marker to represent the last step in a
process flow." name="description" />
        <attribute name="upperName" />
        <attribute value="162" name="y" />
        <attribute value="End Process" name="name" />
        <attribute value="467" name="x" />
      </attributes>
      <typeDef>
        <Reference type="componentVersion"
className="com.netdecisions.components.ccp.steps.DefaultTerminalStep" version="1" />
      </typeDef>
      <repositoryComponentParameters>
        <Reference type="dbo">
          <bo className="com.netdecisions.ccp.dbo.bo.DBO" objectId="1117161310266">
            <objectId>
              <attribute value="1117161310266" name="id" className="java.lang.Long" />
            </objectId>
            <attributes>
              <attribute value="0" name="version" />
            </attributes>
            <values />
            <definition>
              <bo className="com.netdecisions.ccp.dbo.bo.DBODefinition"
objectId="1115971637242">
                <objectId>
                  <attribute value="1115971637242" name="id" className="java.lang.Long" />
                </objectId>
                <attributes>
                  <attribute value="End Process" name="name" />
                  <attribute value="2" name="version" />
                </attributes>
                <valueDefinitions />
              </bo>
            </definition>
          </bo>
        </Reference>
      </repositoryComponentParameters>
      <processVersionFlow>
        <Reference type="link" linkId="16" />
      </processVersionFlow>
      <leadingSteps>
        <Reference type="link" linkId="22" />
        <Reference type="link" linkId="20" />
      </leadingSteps>
    </bo>
    <bo className="com.netdecisions.ccp.model.bp.ConditionStepDefinition" linkId="22">
      <attributes>
        <attribute name="description" />
        <attribute value="179" name="y" />
        <attribute value="1" name="order" />
        <attribute value="367" name="x" />
      </attributes>
      <toStep>
        <Reference type="link" linkId="21" />
      </toStep>
      <fromStep>
        <Reference type="link" linkId="23" />
      </fromStep>
    </bo>
    <bo className="com.netdecisions.ccp.model.bp.StepDefinition" linkId="23">
```

```
      <attributes>
        <attribute value="false" name="isMultiStep" />
        <attribute value="This step is used to stop a process midway through execution."
name="description" />
        <attribute name="upperName" />
        <attribute value="163" name="y" />
        <attribute value="Terminate Process" name="name" />
        <attribute value="220" name="x" />
      </attributes>
      <typeDef>
        <Reference type="componentVersion"
className="com.netdecisions.components.ccp.steps.TerminatingStep" version="1" />
      </typeDef>
      <repositoryComponentParameters>
        <Reference type="dbo">
          <bo className="com.netdecisions.ccp.dbo.bo.DBO" objectId="1115777234746">
            <objectId>
              <attribute value="1115777234746" name="id" className="java.lang.Long" />
            </objectId>
            <attributes>
              <attribute value="0" name="version" />
            </attributes>
            <values />
            <definition>
              <bo className="com.netdecisions.ccp.dbo.bo.DBODefinition"
objectId="1115253289562">
                <objectId>
                  <attribute value="1115253289562" name="id" className="java.lang.Long" />
                </objectId>
                <attributes>
                  <attribute value="Terminate Process" name="name" />
                  <attribute value="2" name="version" />
                </attributes>
                <valueDefinitions />
              </bo>
            </definition>
          </bo>
        </Reference>
      </repositoryComponentParameters>
      <processVersionFlow>
        <Reference type="link" linkId="16" />
      </processVersionFlow>
      <possibleNextSteps>
        <Reference type="link" linkId="22" />
      </possibleNextSteps>
      <leadingSteps>
        <Reference type="link" linkId="24" />
      </leadingSteps>
    </bo>
    <bo className="com.netdecisions.ccp.model.bp.ConditionStepDefinition" linkId="24">
      <attributes>
        <attribute name="description" />
        <attribute value="179" name="y" />
        <attribute value="1" name="order" />
        <attribute value="52" name="x" />
      </attributes>
      <toStep>
        <Reference type="link" linkId="23" />
      </toStep>
      <ruleDefinition>
```

```
        <Reference type="link" linkId="25" />
      </ruleDefinition>
      <fromStep>
        <Reference type="link" linkId="17" />
      </fromStep>
    </bo>
    <bo className="com.netdecisions.ccp.model.rule.RuleDefinition" linkId="25">
      <attributes>
        <attribute value="false" name="isTrueBranch" />
        <attribute value="Determines if a value contains another " name="description" />
        <attribute name="upperName" />
        <attribute value="Contains Text Variable Rule" name="name" />
      </attributes>
      <typeDef>
        <Reference type="componentVersion" className="basic.rules.ContainsTextRule"
version="3" />
      </typeDef>
      <repositoryComponentParameters>
        <Reference type="dbo">
          <bo className="com.netdecisions.ccp.dbo.bo.DBO" objectId="1115001823513">
            <objectId>
              <attribute value="1115001823513" name="id" className="java.lang.Long" />
            </objectId>
            <attributes>
              <attribute value="0" name="version" />
            </attributes>
            <values>
              <bo className="com.netdecisions.ccp.dbo.bo.DBOVariableValue"
objectId="1115745047993">
                <objectId>
                  <attribute value="1115745047993" name="id" className="java.lang.Long" />
                </objectId>
                <attributes>
                  <attribute name="primaryTypeName" />
                  <attribute value="false" name="isNew" />
                  <attribute value="/folder/Trigger/ControlClassName/text()" name="path" />
                  <attribute value="First Var" name="name" />
                </attributes>
              </bo>
              <bo className="com.netdecisions.ccp.dbo.bo.StringDBODataType"
objectId="1115271290201">
                <objectId>
                  <attribute value="1115271290201" name="id" className="java.lang.Long" />
                </objectId>
                <attributes>
                  <attribute name="value" />
                  <attribute value="Constant" name="name" />
                  <attribute value="CallEndedResponse" name="stringValue" />
                </attributes>
              </bo>
              <bo className="com.netdecisions.ccp.dbo.bo.BooleanDBODataType"
objectId="1114650131999">
                <objectId>
                  <attribute value="1114650131999" name="id" className="java.lang.Long" />
                </objectId>
                <attributes>
                  <attribute value="false" name="value" />
                  <attribute value="Case Insensitive Comparison" name="name" />
                </attributes>
              </bo>
```

```
            </values>
            <definition>
              <bo className="com.netdecisions.ccp.dbo.bo.DBODefinition"
objectId="1115546193433">
                <objectId>
                  <attribute value="1115546193433" name="id" className="java.lang.Long" />
                </objectId>
                <attributes>
                  <attribute value="Contains Text Variable Rule" name="name" />
                  <attribute value="2" name="version" />
                </attributes>
                <valueDefinitions>
                  <bo className="com.netdecisions.ccp.dbo.bo.DBOVariableValueDefinition"
objectId="1118235582041">
                    <objectId>
                      <attribute value="1118235582041" name="id" className="java.lang.Long"
/>
                    </objectId>
                    <attributes>
                      <attribute value="false" name="newVariableAllowed" />
                      <attribute name="securityPermissionRequired" />
                      <attribute value="false" name="usesFolderArrayTypes" />
                      <attribute value="true" name="required" />
                      <attribute name="page" />
                      <attribute
value="xsd:integer|xsd:nonPositiveInteger|xsd:negativeInteger|xsd:long|xsd:int|xsd:short|xs
d:byte|xsd:nonNegativeInteger|xsd:unsignedLong|xsd:positiveInteger|xsd:unsignedInt|xsd:unsi
gned_short|xsd:unsignedByte|xsd:float|xsd:double|xsd:decimal|xsd:string|"
name="variableType" />
                      <attribute name="newTypeName" />
                      <attribute value="Compare variable" name="displayText" />
                      <attribute value="false" name="arrayType" />
                      <attribute value="com.netdecisions.ccp.dbo.bo.DBOVariableValue"
name="dataTypeClassName" />
                      <attribute value="First Var" name="name" />
                      <attribute value="false" name="isUsingNewVariable" />
                    </attributes>
                    <defaultValue />
                    <validValues />
                  </bo>
                  <bo className="com.netdecisions.ccp.dbo.bo.DBOVariableValueDefinition"
objectId="1117722651769">
                    <objectId>
                      <attribute value="1117722651769" name="id" className="java.lang.Long"
/>
                    </objectId>
                    <attributes>
                      <attribute value="false" name="newVariableAllowed" />
                      <attribute name="securityPermissionRequired" />
                      <attribute value="false" name="usesFolderArrayTypes" />
                      <attribute value="true" name="required" />
                      <attribute name="page" />
                      <attribute
value="xsd:integer|xsd:nonPositiveInteger|xsd:negativeInteger|xsd:long|xsd:int|xsd:short|xs
d:byte|xsd:nonNegativeInteger|xsd:unsignedLong|xsd:positiveInteger|xsd:unsignedInt|xsd:unsi
gned_short|xsd:unsignedByte|xsd:float|xsd:double|xsd:decimal|xsd:string|"
name="variableType" />
                      <attribute name="newTypeName" />
                      <attribute value="With variable" name="displayText" />
                      <attribute value="false" name="arrayType" />
```

```
                      <attribute value="com.netdecisions.ccp.dbo.bo.DBOVariableValue"
name="dataTypeClassName" />
                      <attribute value="Second Var" name="name" />
                      <attribute value="false" name="isUsingNewVariable" />
                    </attributes>
                    <defaultValue />
                    <validValues />
                  </bo>
                  <bo className="com.netdecisions.ccp.dbo.bo.DBOValueDefinition"
objectId="1118298775161">
                    <objectId>
                     <attribute value="1118298775161" name="id" className="java.lang.Long"
/>
                    </objectId>
                    <attributes>
                      <attribute name="securityPermissionRequired" />
                      <attribute value="true" name="required" />
                      <attribute name="page" />
                      <attribute value="Or with constant" name="displayText" />
                      <attribute value="false" name="arrayType" />
                      <attribute value="com.netdecisions.ccp.dbo.bo.StringDBODataType"
name="dataTypeClassName" />
                      <attribute value="Constant" name="name" />
                    </attributes>
                    <defaultValue />
                    <validValues />
                  </bo>
                  <bo className="com.netdecisions.ccp.dbo.bo.DBOValueDefinition"
objectId="1117954355673">
                    <objectId>
                     <attribute value="1117954355673" name="id" className="java.lang.Long"
/>
                    </objectId>
                    <attributes>
                      <attribute name="securityPermissionRequired" />
                      <attribute value="true" name="required" />
                      <attribute name="page" />
                      <attribute value="Is Comparison Case Insensitive" name="displayText"
/>
                      <attribute value="false" name="arrayType" />
                      <attribute value="com.netdecisions.ccp.dbo.bo.BooleanDBODataType"
name="dataTypeClassName" />
                      <attribute value="Case Insensitive Comparison" name="name" />
                    </attributes>
                    <defaultValue />
                    <validValues />
                  </bo>
                </valueDefinitions>
              </bo>
            </definition>
          </bo>
        </Reference>
      </repositoryComponentParameters>
    </bo>
    <bo className="com.netdecisions.ccp.model.bp.Trigger" linkId="26">
      <attributes>
        <attribute value="Unhandled Control" name="triggerType" />
        <attribute value="true" name="isEnabled" />
        <attribute name="description" />
        <attribute value="Default Unhandled Control Trigger" name="name" />
```

```
      </attributes>
      <processVersionFlow>
        <Reference type="link" linkId="16" />
      </processVersionFlow>
    </bo>
  </Process>
</Processes>
```

B

Additional material

This Redpaper refers to additional material that can be downloaded from the Internet as described below.

Locating the Web material

The Web material associated with this Redpaper is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser to:

ftp://www.redbooks.ibm.com/redbooks/REDP3977

Alternatively, you can go to the IBM Redbooks Web site at:

ibm.com/redbooks

Using the Web material

The additional Web material (REDP3977.zip) that accompanies this Redpaper includes the files in Table B-1.

Table B-1 Additional material

File name or directory	Description
pkginfo.txt	UNIX text file containing the output from the **pkginfo** command

System requirements for downloading the Web material

The following system configuration is recommended:

Hard disk space: Any
Operating System: Windows 2000
Processor: Any
Memory: Any

© Copyright IBM Corp. 2005. All rights reserved.

How to use the Web material

Create a subdirectory (folder) on your workstation, for example, C:\ITSO\REDP3977, and extract REDP3977.zip file into this directory.

Glossary

ANI. See Automatic Number Identification.

ASR. See Automatic Speech Recognition.

Automatic Number Identification (ANI). A service that tells the recipient of a telephone call the telephone number of the person making the call.

Automatic Speech Recognition (ASR). Speech recognition technologies that allow computers equipped with a source of sound input, such as a telephone handset or microphone, to interpret human speech.

barge-in. A capability that allows callers to speak or enter their responses during the prompt.

Call Control eXtensible Markup Language (CCXML). An XML standard designed to provide telephony support to VoiceXML. Where VoiceXML is designed to provide a VUI interface to a voice browser, CCXML is designed to inform the voice browser how to handle the telephony control of the voice channel. The two languages are wholly separate are not required by each other to be implemented.

call flow. The "path" that a caller is directed through when interacting with a voice response system.

CCXML. See Call Control eXtensible Markup Language.

cluster. A group of locally connected computers that work together as a unit.

component. A predefined segment of a voice-driven application.

Computer Telephony Integration (CTI). The name given to the merger of traditional telecommunications (PBX) equipment with computers and computer applications. The use of caller ID to retrieve client information automatically from a database is an example of a CTI application.

confidence score. A value that indicates how certain the recognizer is that the result is correct. The confidence value can be used by the application to decide whether to proceed without asking for confirmation of the data, whether to ask for confirmation of the data, or whether to ask the caller to repeat the data.

configuration. The arrangement of the software and hardware in a computer system or network.

Configurator. The SAB Graphical User Interface (GUI).

confirmation. Short dialog flow that repeats a recognition result back to the caller and asking him or her to confirm that it is correct.

CTI. See Computer Telephony Integration

database. A collection of information stored in a computer in a systematic way, so that a computer program can consult it to answer questions.

debug. The process of locating and correcting errors in computer programs.

default. A particular setting or value for a variable that is assigned automatically by an operating system and remains in effect unless canceled or overridden by the operator.

Dialed Number Identification Service (DNIS). A telephone function that sends the dialed telephone number to the answering service.

dialog box. A secondary window in which you are asked to fill in information.

dialog. Interactions between the caller and the application are called dialogs.

directory. A type of file used to group and organize other files or directories. Also called a folder.

DNIS. See Dialed Number Identification Service.

DTMF. See Dual Tone Multi Frequency tones.

Dual Tone Multi Frequency tones (DTMF). The system used by touch-tone telephones. DTMF assigns a specific frequency (made up of two separate tones) to each key so that it can easily be identified by a microprocessor. This is basically the technology behind touch tone dialing.

form. A document (printed or electronic) with spaces in which to write or enter data.

Graphical User Interface (GUI). A user interface based on graphics (icons and pictures and menus) instead of text.

GUI. See Graphical User Interface.

hardware. The physical components of a computer system, such as keyboards, monitors, and media disk drives.

© Copyright IBM Corp. 2005. All rights reserved.

Integrated Services Digital Network (ISDN). An international standard for end-to-end digital transmission of voice, data, and signaling.

interactive Voice Response (IVR). A telecommunications system that uses prerecorded voice messages to present options to a user.

interface. The point of interaction or communication between a computer and any other entity, such as a printer or human operator.

ISDN. See Integrated Services Digital Network.

IVR. See Interactive Voice Response.

J2EE. See Java 2 Platform, Enterprise Edition.

Java 2 Platform, Enterprise Edition (J2EE). An environment for developing and deploying enterprise applications, defined by Sun Microsystems™ Inc. The J2EE platform consists of a set of services, application programming interfaces (APIs), and protocols that provide the functionality for developing multi-tiered, Web-based applications.

Media Resource Control Protocol (MRCP). The Media Resource Control Protocol created by the Internet Engineering Task Force (IETF). It is a communication protocol that allows speech servers to provide various speech services (such as speech recognition and speech synthesis) to its clients. Typically, this means the server software will be running on one computer and clients can send MRCP messages to the server over a network, usually on top of another protocol, such as RTSP or TCP.

menu. A graphical user interface element that allows the user to select one among several, presumably related, choices.

module. A separate and distinct unit of hardware or software that may be used as a component in a system.

MRCP. See Media Resource Control Protocol.

Natural Language Speech Recognition (NLSR). An advanced type of speech recognition. NLSR can recognize particular words and phrases spoken by the user.

NLSR. See Natural Language Speech Recognition.

node. A device connected to a network, such as a computer or router.

parameter. A value or object that is used to transfer information to or from subprograms. Input parameters can be thought of as being copied into the subprogram when it is called, and output parameters can be thought of as having their values copied back to the caller when the subprogram returns.

PBX. See Private Branch eXchange.

platform. A platform refers to a framework on which applications may be run.

port. A connection to a device that allows data or information to travel into and out of the device.

predictive dialing. An automated dialing feature in which the systems predicts from pre-set algorithms when an agent/operator will become free and makes the call in advance.

Private Branch eXchange (PBX). A private telephone switching system, usually located on a company's premises.

prompt. A message played to a caller that gives the caller a choice of selections in a menu and asks for a response. Compare to announcement.

proxy server. A server that receives requests intended for another server and that acts on the behalf of the client (as the client proxy) to obtain the requested service. A proxy server is often used when the client and the server are incompatible for direct connection.

PSTN. See Public Switched Telephone Network.

Public Switched Telephone Network (PSTN). The traditional analog network that routes voice calls from one location to another.

query. A query is a form of questioning in SQL language to retrieve information from a database.

RDBMS. See Relational Database Management System.

Real Time Streaming Protocol (RTSP). A client/server communication protocol that simplifies the distribution of multimedia contents on the Internet.

Red Hat. A software company in the business of assembling open source components for the Linux operating system and related programs into a distribution package.

Relational Database Management System (RDBMS). A type of DBMS in which the database is organized and accessed according to the relationships between data values.

resilience. The ability of a structure to sustain the impact of a business interruption and recover and resume its operations to continue to provide minimum services.

round robin. A simple mechanism used by DNS servers to share and distribute loads for network resources.

RTSP. See Real Time Streaming Protocol.

scalability. The ability to increase or decrease size or capability in cost-effective increments with minimal impact on the unit cost of business and the procurement of additional services.

script. The set of instructions for the target voice response system to follow during a transaction.

servlet. A server-side Java program that provides additional features to the server.

software. The set or sets of programs that instruct the computer hardware to perform a task or series of tasks.

Speech Synthesis Markup Language (SSML). A markup language to control speech synthesis and text processing defined by the World Wide Web Consortium (W3C).

SSML. See Speech Synthesis Markup Language.

SUSE. A privately owned company whose mission is to promote open source development and General Public License distribution and to be a Linux distribution provider. SUSE assembles open source components for the Linux operating system and related programs into a selection of distribution packages.

Text-to-Speech (TTS). An optional feature that lets an application play language speech directly from ASCII text by converting that text to synthesized speech. The text can be used for prompts or for text retrieved from a database or host, and can be spoken in an application with prerecorded speech.

thin client. A low-cost computing device that accesses applications and data from a central server over a network. Categories of thin clients include Windows-Based Terminals (WBT, which comprise the largest segment), X-Terminals, and Network Computers (NC).

topology. The physical layout of a network.

troubleshooting. The process of locating and correcting errors in computer programs.

TTS. See Text-to-Speech.

turnkey. A product or system that can be plugged in, turned on, and operated with little or no additional configuring.

Uniform Resource Identifier (URI). A URI is a formatted string used to locate, via name or location, a resource document, such as a VoiceXML document.

URI. See Uniform Resource Identifier

Voice eXtensible Markup Language (VoiceXML). The W3C's standard XML format for specifying interactive voice dialogues between a human and a computer. It is fully analogous to HTML, and brings the same advantages of Web application development and deployment to voice applications that HTML brings to visual applications.

VoiceXML. See Voice eXtensible Markup Language.

WAR. See Web ARchive.

Web ARchive (WAR). A Web ARchive file is the new J2EE standard for packaging Java Servlet based application, Java Server Pages, and associated files and deploying them.

Abbreviations and acronyms

ANI	Automatic Number Identification
ASR	Automatic Speech Recognition
CCXML	Call Control Extensible Markup Language
CTI	Computer Telephony Integration
DNIS	Dialed Number Identification Service
DTMF	Dual Tone Multi Frequency
GUI	Graphical User Interface
IBM	International Business Machines Corporation
ISDN	Integrated Services Digital Network
ITSO	International Technical Support Organization
J2EE	Java 2 Platform, Enterprise Edition
MRCP	Media Resource Control Protocol
NLSR	Natural Language Speech Recognition
PBX	Private Branch Exchange
PSTN	Public Switched Telephone Network
RDBMS	Relational Database Management System
RTSP	Real Time Streaming Protocol
SSML	Speech Synthesis Markup Language
TTS	Text-to-Speech
URI	Uniform Resource Identifier
VoiceXML	Voice Extensible Markup Language

© Copyright IBM Corp. 2005. All rights reserved.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this Redpaper.

IBM Redbooks

For information about ordering these publications, see “How to get IBM Redbooks” on page 121. Note that some of the documents referenced here may be available in softcopy only.

- *IBM WebSphere Application Server V5.1 System Management and Configuration WebSphere Handbook Series*, SG24-6195
- *IBM WebSphere V5.1 Performance, Scalability, and High Availability WebSphere Handbook Series*, SG24-6198
- *WebSphere Voice Server for Multiplatforms V5.1.1/V5.1.2 Handbook,* SG24-6447

Other publications

These publications are also relevant as further information sources:

- *Avaya Interactive Response Release 1.3 MRCP Feature*
- *Avaya Speech Application Builder's Configuration and Deployment Guide*
- *Avaya Interactive Response Installation, Migration and Troubleshooting Guide*

Online resources

These Web sites and URLs are also relevant as further information sources:

- Avaya documentation

  http://www.avayadocs.com
- Avaya support page

  http://support.avaya.com
- Cleo home page

  http://www.cleo.com
- Gold Systems home page

  http://www.goldsystems.com
- Microsoft Window Server 2003 TechCenter

  http://www.microsoft.com/technet/prodtechnol/windowsserver2003/default.mspx
- NMS Communications

  http://www.nmscommunications.com
- Real Soft Inc. home page

  http://www.realsoftinc.com

© Copyright IBM Corp. 2005. All rights reserved.

- Red Hat Enterprise Linux server

  http://www.redhat.com/software/rhel
- Softel home page

  http://www.softel.com
- Sun Blade 150 Workstation Overview

  http://www.sun.com/desktop/workstation/sunblade150
- Sun Blade V240 Server Overview

  http://www.sun.com/servers/entry/v240
- SUSE LINUX Enterprise Server

  http://www.novell.com/products/linuxenterpriseserver
- World Wide Web Consortium (W3C) home page

  http://www.w3c.org
- W3C recommendation of Speech Recognition Grammar Specification (SRGS) V1.0

  http://www.w3.org/TR/2004/REC-speech-grammar-20040316
- W3C recommendation of Speech Synthesis Markup Language (SSML) V1.0

  http://www.w3.org/TR/2004/REC-speech-synthesis-20040907
- W3C working draft of Semantic Interpretation for Speech Recognition (SISR)

  http://www.w3.org/TR/2003/WD-semantic-interpretation-20030401
- W3C recommendation for Voice Extensible Markup Language (VoiceXML) V2.0

  http://www.w3.org/TR/voicexml20
- WebSphere Application Server Zone

  http://www.ibm.com/developerworks/websphere/zones/was
- WebSphere Application Server V5.1.x Information Center

  http://publib.boulder.ibm.com/infocenter/ws51help/index.jsp?topic=/com.ibm.websphere.base.doc/info/welcome_base.html
- WebSphere Application Server Network Deployment V5.1.x Information Center

  http://publib.boulder.ibm.com/infocenter/ws51help/index.jsp?topic=/com.ibm.websphere.nd.doc/info/welcome_nd.html
- WebSphere Application Server Base and Network Deployment editions recommended updates

  http://www.ibm.com/support/docview.wss?rs=180&uid=swg27004980
- *WebSphere Edge Server for Multiplatforms Network Dispatcher (Load Balancer) Administration Guide Version 2.0*, GC31-8496, found at:

  ftp://ftp.software.ibm.com/software/websphere/edgeserver/info/doc/v20/en/NDguide.pdf
- WebSphere Voice family product documentation library

  http://www.ibm.com/developerworks/websphere/zones/voice/proddoc.html#wvs
- WebSphere Voice Server Education and Class Information

  http://www.ibm.com/developerworks/websphere/education/enablement
- WebSphere Voice Server for Multiplatforms V5.1.x Information Center

  http://publib.boulder.ibm.com/infocenter/pvcvoice/51x/index.jsp
- WebSphere Voice Server Support Page

  http://www-306.ibm.com/software/pervasive/voice_server/support

- WebSphere Voice Server System Requirements

  http://www.ibm.com/software/pervasive/voice_server/system_requirements

- WebSphere Voice Server IVR and gateway compatibility Web page

  http://www.ibm.com/software/pervasive/voice_server/ivrgateway.html

- WebSphere Voice Toolkit

  http://www.ibm.com/software/pervasive/voice_toolkit
  http://www3.software.ibm.com/ibmdl/pub/software/websphere/voicetoolkit

- WebSphere Voice Zone

  http://www.ibm.com/developerworks/websphere/zones/voice

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

IBM WebSphere Voice Server V5.1.1/V5.1.2 and Avaya Interactive Response V1.3: An Interoperability Guide

Redpaper

Use Media Resource Control Protocol (MRCP) for interoperability

Apply best practices to VoiceXML development

Test and debug VoiceXML applications

This IBM Redpaper is an interoperability guide for integrating WebSphere Voice Server V5.1.1/V5.1.2 and Avaya Interactive Response V1.3. We give you a broad understanding of how open standards like VoiceXML, Media Resource Control Protocol (MRCP), Speech Recognition Grammar Specification (SRGS), and Speech Synthesis Markup Language (SSML) are used.

In this Redpaper, we consider the best practices applied to Voice User Interface (VUI) design. We implement and deploy a simple voice-enabled application using Avaya Interactive Response that will interoperate with WebSphere Voice Server V5.1.1/V5.1.2.

We demonstrate how to develop, test, and deploy a simple voice-enabled application using the VoiceXML markup language. We also demonstrate the use of Automatic Speech Recognition (ASR) and Text-to-Speech (TTS) voice technologies through examples.

With this Redpaper, you can tailor and configure WebSphere Voice Server and Avaya Interactive Response to interoperate in simple and complex topologies.

We assume you have a basic knowledge of Interactive Voice Response (IVR) systems and voice-enablement of applications using VoiceXML, Avaya Interactive Response, WebSphere Voice Server, and WebSphere Application Server.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks